Project report

ON

ROBOTIC ARM CONTOL USING HAPTIC GLOVE

Submitted by:

SAI ANIRUDH B 01FB15EEC193

RAKSHITH NARUN 01FB16EEC718

HARISH KUMAR S 01FB16EEC706

Under the guidance of

Guide
SUMANTH SAKKARA
ASST.PROFESSOR, ECE
PESU

January – May 2019

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

FACULTY OF ENGINEERING
"PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560085, Karnataka, India

CERTIFICATE

This is to certify that the Report entitled

ROBOTIC ARM CONTOL USING HAPTIC GLOVE

is a bonafide work carried out by

SAI ANIRUDH B 01FB15EEC193

RAKSHITH NARUN 01FB16EEC718

HARISH KUMAR S 01FB16EEC706

In partial fulfillment for the completion of 8th semester course work in the Program of Study B.Tech in Electronics and Communication Engineering, under rules and regulations of PES University, Bengaluru during the period Jan – Apr. 2019. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 8th semester academic requirements in respect of project work.

Signature with date & Seal	*Signature with date & Seal*	*Signature with date & Seal*
Asst.Prof.Sumanth SAKKARA	*Dr. Anuradha M*	*Dr. B. K. Keshavan*
Internal Guide	*Chairperson*	*Dean - Faculty of Engg. &Technology*

Dr.D.Sethuram

Co - Guide

Name and signature of the examiners:

1.

2.

3.

DECLARATION

We hereby declare that the project entitled **"ROBOTIC ARM CONTROL USING HAPTIC GLOVE"** has been carried out by us under the guidance of SUMANTH SAKKARA, Professor, PESU and submitted in fulfillment of the course requirements for the award of degree of **Bachelor of Technology** in **ELECTRONICS AND COMMUNICATION ENGINEERING** of **PES University, Bengaluru** during the academic semester January – May 2019. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

Signature

SAI ANIRUDH B 01FB15EEC193

RAKSHITH NARUN 01FB16EEC718

HARISH KUMAR S 01FB16EEC706

ACKNOWLEDGEMENT

We would like to take this opportunity to give our sincerest gratitude to Sumanth Sakkara and Dr.D.Sethuram(mechanical Dept), for giving us the inspiration, guidance and wisdom to carry out this project. This project would have not been completed without their enormous help and worthy experience.

We would like to thank our project coordinators for providing the required information and updates at the appropriate times and aiding us through the entire process.

We would like to extend our heartfelt thanks to Dr Anuradha, Head Of Department , Department of ECE for not only giving us this opportunity but for supporting all the activities that is associated with it.

We would also like to give a special thanks to Dr BK Keshavan, Dean of Faculty for his support throughout the process.

We would like also like to thank Dr K N B Murthy, Vice Chancellor PES Institutions for his continuous encouragement and motivation.

We would also like to extend our sincerest gratitude to Prof Jawahar Doreswamy Pro Chancellor PES Institutions for his inspirational dedication and support to all activities associated with PES.

We would like to thank Dr M R Doreswamy Chancellor PES Institutions for his enormous support and motivation

Finally, we would like to thank our parents who have always supported us and encouraged us to do better and providing us with constant care and affection.

CONTENTS

LIST OF ABBREVIATIONS

MEMS - Micro Electro-Mechanical Systems

FDA- Fused Deposition Modelling

3DP: 3D Printing

ABS: Acrylonitrile Butadiene Styrene

AM: Additive Manufacturing

CAD: Computer-Aided Design

CAM: Computer-Aided Manufacturing

WI-FI: wireless fidelity

SLA: Stereo lithography Apparatus

SLS: Selective Laser Sintering

STL/.stl: Stereo Lithographic

PLA-Polylactic acid

DIP - Distal Interphalangeal

PIP - Proximal Interphalangeal

MCP - Metacarpophalangeal”

Glossary

3D Printing: The process where a solid object is created from a computer model. The term 3D printing is often used interchangeably with additive manufacturing. Parts are made in plastic, rubber, or metal.

Bed: Another name for a 3D printer build plate. This is the flat surface in which the parts are made.

Fused Deposition Modeling (FDM): 3D printing process where melted plastic material is extruded from a nozzle, and layers of material are extruded to build parts. Both the term and its acronym are trademarks of Stratasys.

Filament: This is the wire made from build material which enters the cold end of the extruder, or the heated wire which exits the hot end of the extruder.

List of Figures

CHAPTER 1: INTRODUCTION

1.1 Introduction:

The project of Robotic Arm controlled via Haptic Glove is a system by which a user can control a remotely placed robotic arm via the haptic glove controller. We are including a way by which the user shall send the angle of each finger to the arm via the google firebase cloud. Google Firebase cloud serves as a real time database where each time the data changes it is fed to the cloud via the glove's esp8266 module which we have preloaded with the security keys ,access url of the database in order to make it secure. This access key and the secret key is also shared with the robotic arm by which it receives this data and moves the robotic arm accordingly. The focus of this project was to make a robotic arm which can make almost all the gestures which a normal arm could possibly make, to name one of the custom motions we have developed is an abduction mechanism which allows the user to flex his Intrinsic muscles Lumbricals and interossei which allows the following motion.

Fig.1.1

The haptic glove is equipped with a flex sensor as well as accelerometers this is done as polling of the accelerometer can be avoided. This is a much costly affair compared to just pooling for the angle when we notice a flex (resistance) has changed.

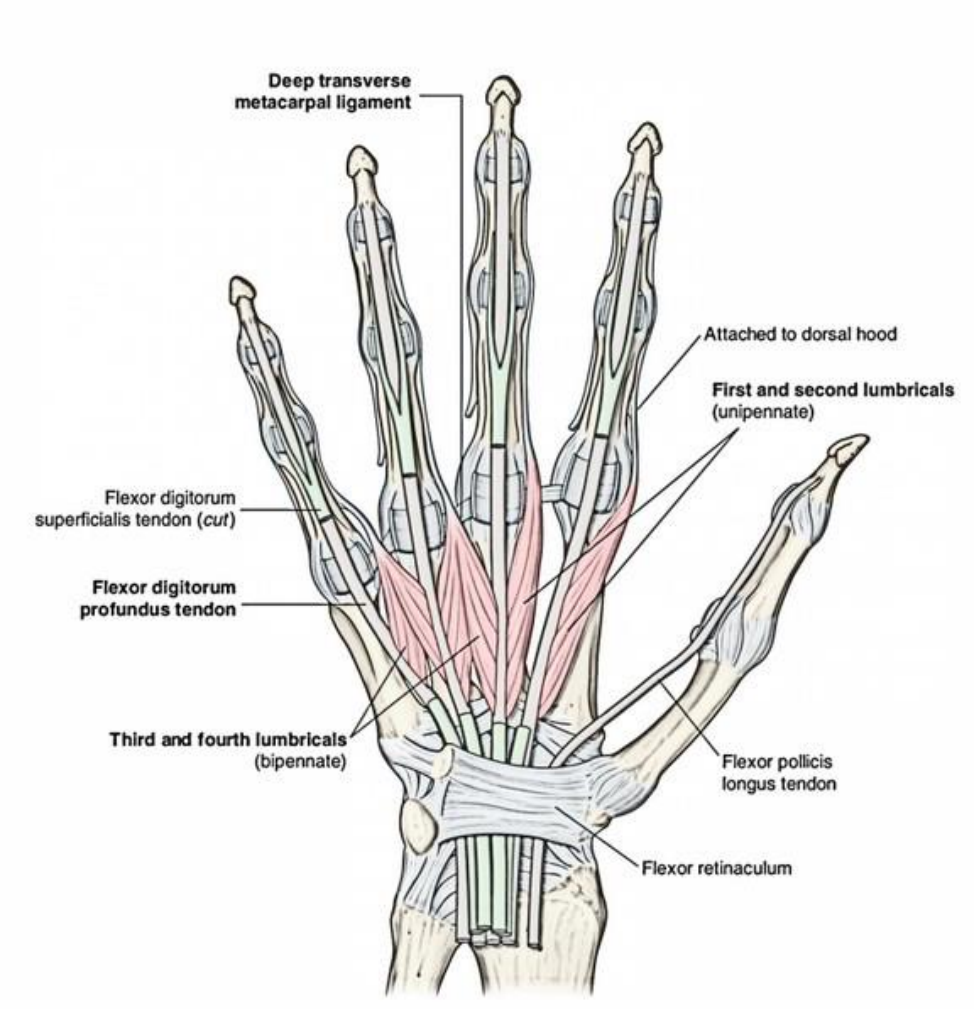

Fig.1.2

1.2 Problem Definition:

Robotics have become an essential part of today's world. Almost all the products that we are familiar with have come in contact with robots one way or another. Prosthetics is another area which is becoming more and more researched into as now a prosthetic not necessarily means a replacement for something you lost but replacement for something better. As in a human hand can only hold objects which are in the temperature range of 20°C - 50°C comfortably, beyond which it may cause irritation and pain. But this is not a limitation for a robotic arm which can hold the object as long as the material which is used in the making of the robot can tolerate which will be at least 5x times that of a human arm.

Also many of the prosthetics since designed for strength are not human like but are purposed differently. Our aim in this project is to create a robotic arm which is capable of

performing complex gestures which are possible only by a human arm. Since the field of use of the arm is unknown to us, we have to design considering the worst case that the arm and the glove are in 2 ends of the world but should still be able to communicate with each other.

1.3 Objective of the Project:

As a beneficial solution to the aforementioned issue, certain factors are required to be considered before designing a well stabilized arm and glove.

- **Scalability** – This refers to the scenario where there could be multiple robotic arms and gloves. The communication between them must be perfected so that they do not control/get controlled by another arm/glove.
- **Reliability** – Being a safe and secure system with features to make sure that no damage could happen to the arm because of the force/temperature exerted on it despite the controller of the arm physically not being present there/arm is not visible to him.
- **Cost** – To address the issue of lack of reliability in a cheaper solution to prosthetics system generally used, the solution must be capable of proving to be for a custom 3d printed model cost effective.
- **Maintenance** – Considering that the area of application is to be generic and thus even for organizations with low resources, the system must require less maintenance which can be easily done as required. Components of the prosthetic must be easily changeable in case of damage to only part of the existing robotic arm.
- **Ease of use** – As this system could be implemented in any field, the system should be easy to use by any person, with any level of education or training. The system must not be too complex for an authorized person to use.

Taking all these factors into consideration, the solution proposed is a custom 3-D printed **Robotic arm which is controlled by a haptic glove.**

Similar to already available robotic arms, various sizes/temperature of objects can be held. Each finger would have its own sensors to read the angle so one each finger of the

robotic arm could be controlled from the haptic glove's side. Since the linking of the controller to the arm is happening via an online database any 3rd party authorized person can see the status of both the devices, for their working. This make the implementation of the project not only scalable but highly secure as well.

The information could be accessed with any computer connected to the network. The information could also be maintained over the cloud based on the organization's resources and requirements. The system would thus be reliable depending on the method of implementation chosen.

The robotic arm is designed top to bottom in-house which make the printing of the custom robotic arm and printing it very affordable when compared to other available options in the market. This would have been the most expensive component in our project. But this would be only a one time investment, as you will not have to buy the entire arm again and again. The long term maintenance required /replacement in case of any damage done to a part. Only those parts can also be easily replaced as each of this item can be 3d printed and fitted back without much design/cost. Thus, the system would be cost effective/easy to maintain.

Once the system is setup with each other the control of the robotic arm via the haptic glove is extremely simple. As long as the user moves his palm, the robotic arm shall also mimic the same.

CHAPTER 2: LITERATURE SURVEY

1 ROBOTICS ARM CONTROL USING HAPTIC TECHNOLOGY

Vipul J. Gohil, Dr. S D. Bhagwat, Amey P. Raut3, Prateek R. Nirmal have worked on "ROBOTICS ARM CONTROL USING HAPTIC TECHNOLOGY" and presented it in International Journal of Latest Research in Science and TechnologyVolume 2,Issue 2 :Page No.98-102 , March - April (2013). Robots of the current generation have been used in fields isolated from the human society. The main objective of the project is to design and develop the Robot that is used to move using wireless system by recognizing hand motion that is controlled by haptics technology for virtual environment & human-machine systems capable of haptic interaction. The proposed system is utilized to recognize the human motion..Large potential for applications in critical fields as well as for leisurely pleasures. Haptic devices must be smaller so that they are lighter, simpler and easier to use.Haptic technology allows interactivity in real-time with virtual objects.

2 Flex Sensor Based Robotic Arm Controller Using Micro Controller

Abidhusain Syed, Zamrrud Taj H. Agasbal, Thimmannagouday Melligeri, Bheemesh Gudur worked on "Flex Sensor Based Robotic Arm Controller Using Micro Controller" " and presented it in Department of Electronics and Communication, BLDEA College of Enggineering Bijapur-3, India, JSEA> Vol.5 No.5, May 2012. Sensor plays an important role in robotics. Sensors are used to determine the current state of the system. Robotic applications demand sensors with high degrees of repeatability, precision, and reliability. Flex sensor is such a device, which accomplish the above task with great degree of accuracy. The pick and place operation of the robotics arm can be efficiently controlled using micro controller programming. The paper discussed a hardware and software co design of robotic arm controller using four servomotors employing micro controller. Micro controller programming can be done with an ease to suit the requirements. Unlike which employ FPGA based control. Micro controller based programs can be flexibly modified to suit the necessary drive control of the serve motor.

3 Design of a Haptic Arm Exoskeleton for Training and Rehabilitation

Abhishek Gupta, Student Member, IEEE, and Marcia K. O'Malley, Member, IEEE worked on "Design of a Haptic Arm Exoskeleton for Training and Rehabilitation". In this paper, the authors present a detailed review of the requirements and constraints that are involved in the design of a high-quality haptic arm exoskeleton. In this context, the design of a five-degree-of-freedom haptic arm exoskeleton for training and rehabilitation in virtual environments is presented. This paper presents the first iteration of the design of a haptic arm exoskeleton for rehabilitation and training. The workspace of the robot encompasses almost 90%ofthetotalhumanforearmworkspace, except for the limitation in the flexion of the elbow joint. There exist no singularities in the workspace of the robot. The arm-centred design results in a compact interface that does not compromise natural arm movements. The alignment of human and robot axes permits easy measurement of human arm joint angles along with increased control over independent feedback to individual human arm joints.

4 Gesture Control Robotic Arm Using Flex Sensor

Waseem Afzal, Shamas Iqbal, Zanib Tahira, Mehtab Ejaz Qureshi worked on"Gesture Control Robotic Arm Using Flex Sensor". The design and implementation of a gesture control robotic arm using flex sensor is proposed. The robotic arm is designed in such a way that it consists of four movable fingers, each with three linkages, an opposing thumb, a rotating wrist and an elbow. The robotic arm is made to imitate the human hand movements using a hand glove. The hand glove consists 5 flex sensor for controlling the finger movements and an Accelerometer for the wrist and elbow movements. The actuators used for the robotic arm are servo motors. The finger movements are controlled using cables that act like the tendons of human arm. The robotic arm is controlled from a distant location using a wireless module. A prototype of the robotic arm was constructed and tested for various hand movements. The robotic arm was made of low cost materials that were readily available. The model of the robotic arm was constructed and the functionality was tested.

CHAPTER 3: METHODOLOGY AND IMPLEMENTATION

3. 1 Project Flow

The basic outer algorithm of the entire system can be perceived from the flowchart below:

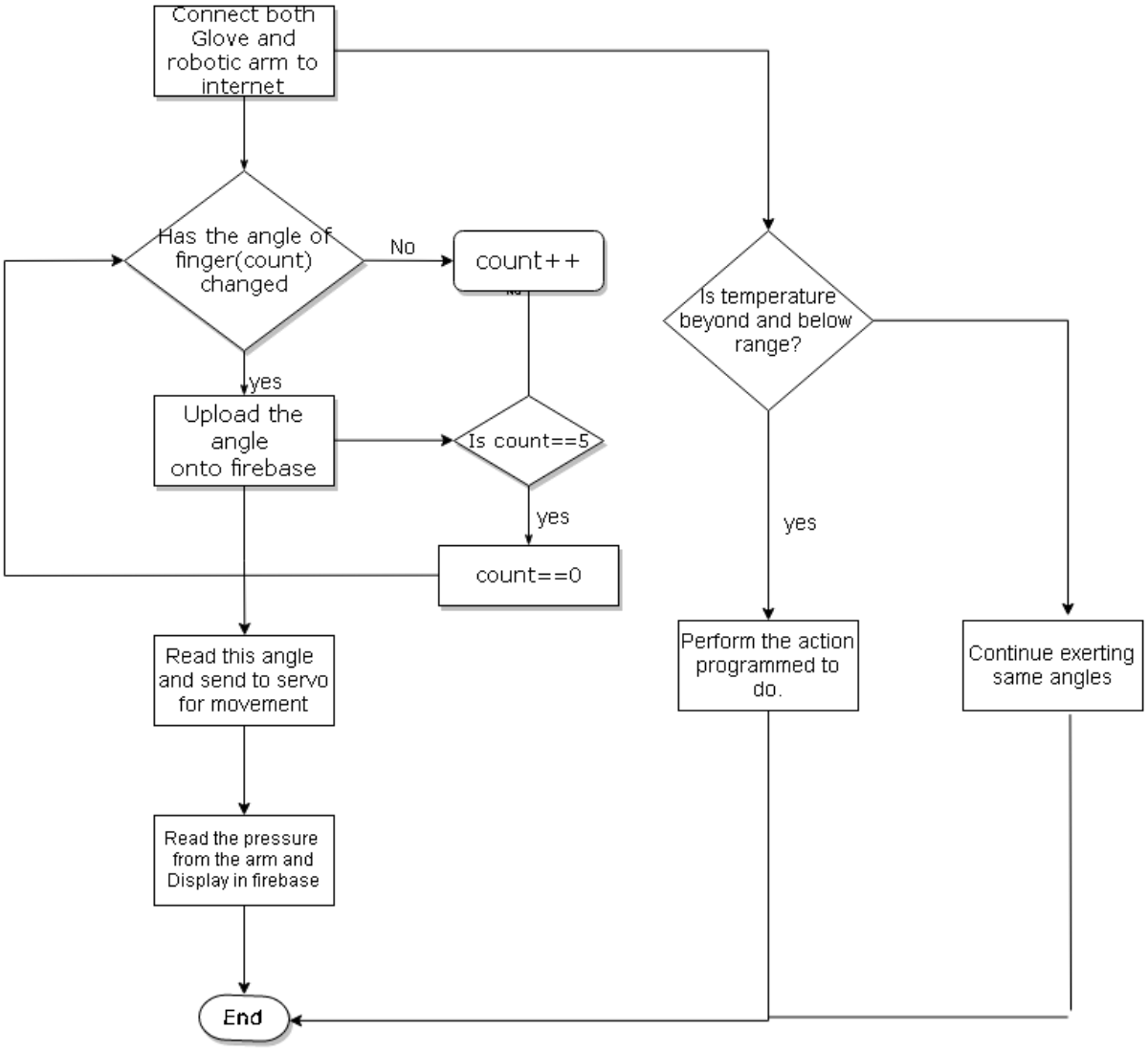

Fig.3.1 Flow Chart

The project starts off with the glove reading the angles of the 5 fingers in a round robin fashion and uploading all of these angles to the firebase database.

The robotic arm on the other end after retrieving these angles has to manipulate the servos in order for in to reach the same angle on the arm as well. This process will be happening continuously throughout the arm is being controlled.

Another section will be the feedback from the arm to the robotic glove. This feedback includes the temperature of the object that the user is holding onto and the pressure with which he is holding the object. Both of these also are uploaded onto the firebase, so we will have a dashboard where the user can keep monitoring these parameters and decide on what action he would like to take.

The safety feature kicks in when the object being held can be damaging to the arm. This feature can easily be overridden.

3.2 Hardware Architecture and Design

3.2.1 Components Description

1 Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous Programmable circuit boards, the Arduino does not need a separate pieceof hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

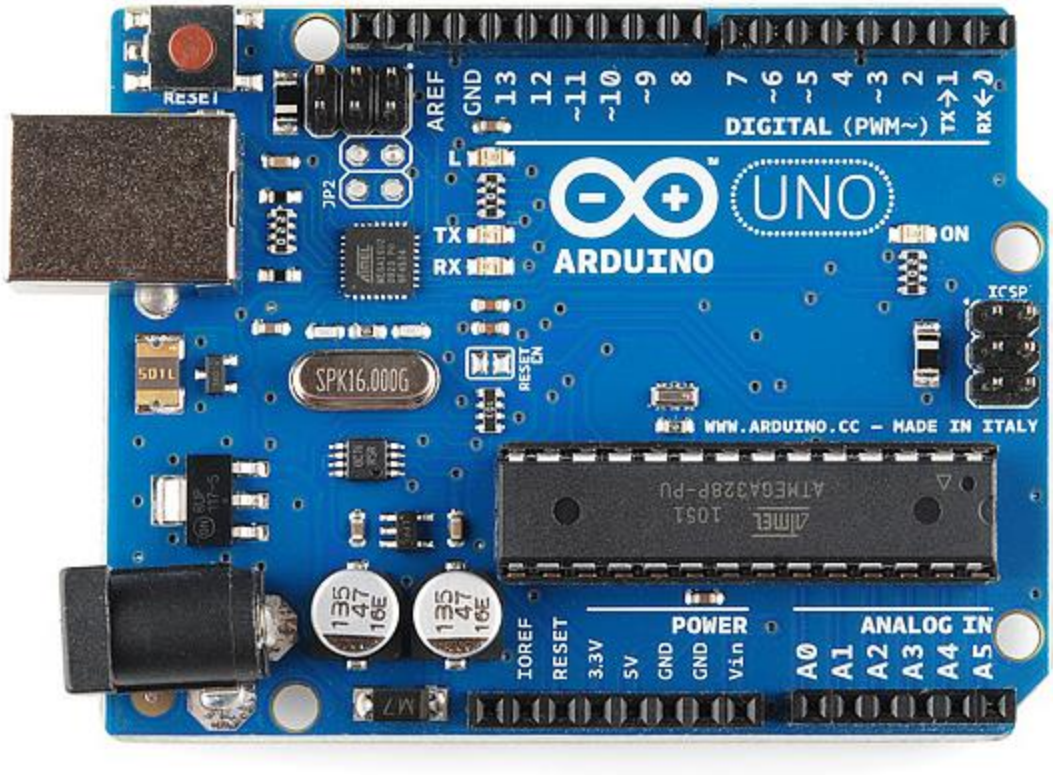

Fig.3.2 Arduino Uno

2 Accelerometer(ADXL-345)

The ADXL345 is a low-power, 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The Adafruit Breakout boards for these modules feature on-board 3.3v voltage regulation and level shifting which makes them simple to interface with 5v microcontrollers such as The ADXL345 features 4 sensitivity ranges from +/- 2G to +/- 16G. And it supports output data rates ranging from 10Hz to 3200Hz. The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against forces due to applied acceleration.

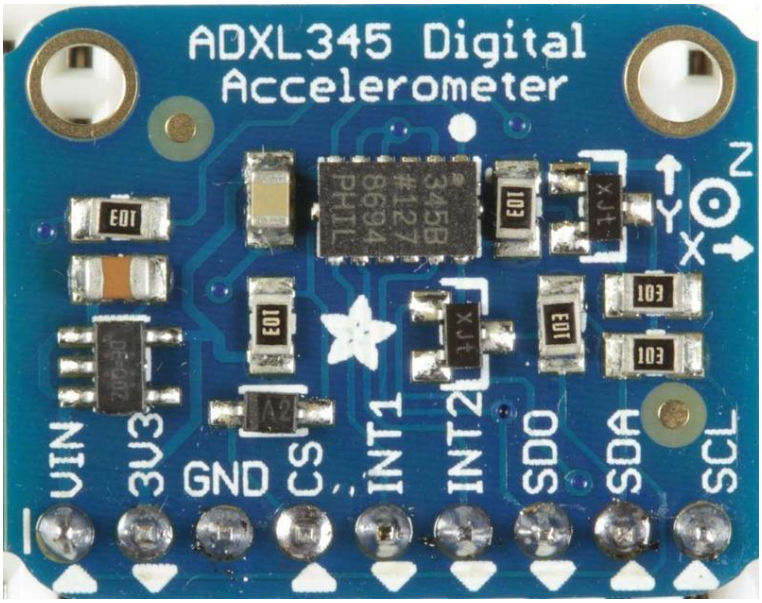

Fig.3.3 ADXL-345

3 Servos

A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped). This type of servomotor is not widely used in industrial motion control, but it forms the basis of the simple and cheap servos used for radio-controlled models.

Fig.3.4 Servo

4 Temperature Sensor

Temperature measurement can be enabled or skipped. Skipping the measurement could be useful to measure pressure extremely rapidly. When enabled, several oversampling options exist. Each oversampling step reduces noise and increases the output resolution by one bit, which is stored in the XLSB data register 0xFC. Enabling/disabling the temperature measurement and oversampling setting are selected through the osrs_t[2:0] bits in control register 0xF4.

Sleep mode Sleep mode is set by default after power on reset. In sleep mode, no measurements are performed and power consumption (IDDSM) is at a minimum. All registers are accessible; Chip-ID and compensation coefficients can be read.

Forced mode In forced mode, a single measurement is performed according to selected measurement and filter options. When the measurement is finished, the sensor returns to sleep mode and the measurement results can be obtained from the data registers. For a next measurement, forced mode needs to be selected again. This is similar to BMP180 operation. Forced mode is recommended for applications which require low sampling rate or host-based synchronization.

Normal mode Normal mode continuously cycles between an (active) measurement period and an (inactive) standby period, whose time is defined by standby. The current in the standby period (IDDSB) is slightly higher than in sleep mode. After setting the mode, measurement and filter options, the last measurement results can be obtained from the data registers without the need of further write accesses. Normal mode is recommended when using the IIR filter,

and useful for applications in which short-term disturbances (e.g. blowing into the sensor) should be filtered.

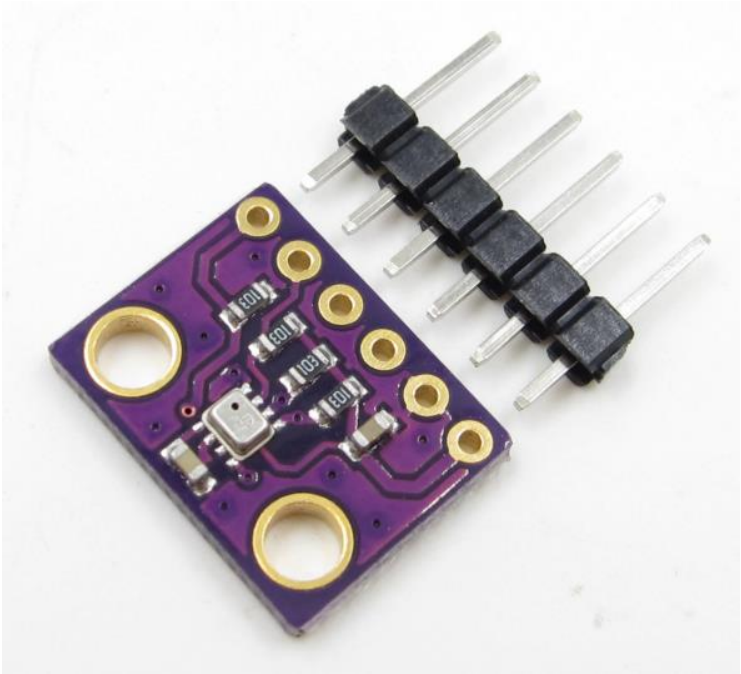

Fig.3.5 Temperature Sensor

5 FSR Sensor

FSRs are sensors that allow you to detect physical pressure, squeezing and the weight being exerted on it.

They are simple to use and low cost. This is a photo of an FSR, specifically the Interlink 402 model. The 1/2" diameter round part is the sensitive bit.

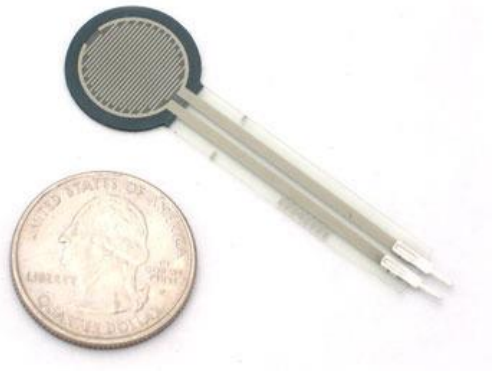

Fig.3.6 FSR Sensor

The FSR is made of 2 layers separated by a spacer. The more one presses,the more of those Active Element dots touch the semiconductor and that makes the resistance go down.

FSRs are basically a resistor that changes its resistive value (in ohms Ω) depending on how much it is pressed. These sensors are fairly low cost, and easy to use but they're rarely accurate. They also vary some from sensor to sensor perhaps 10%. So basically when you use FSRs you should only expect to get *ranges* of response. While FSRs can detect weight, they're a bad choice for detecting exactly how many pounds of weight are on them. However, for most touchsensitive applications like "has this been squeezed or pushed and about how much" they're a good deal for the money!
Some Basic StatsThese stats are specifically for the Interlink 402, but nearly all FSRs will be similar. Checking the datasheet will always illuminate any differences

- **Size:** 1/2" (12.5mm) diameter active area by 0.02" thick (Interlink does have some that are as large as 1.5"x1.5")
- **Price**-Rs.600
- **Resistance range:** Infinite/open circuit (no pressure), 100KΩ (light pressure) to 200Ω (max. pressure)
- **Force range**: 0 to 20 lb. (0 to 100 Newtons) applied evenly over the 0.125 sq in surface area
- **Power supply:** Any! Uses less than 1mA of current (depends on any pullup/down resistors used and supply voltage)
- **Datasheet** (note there are some mathematical inconsistencies in here)

How to measure force/pressure with an FSR

As we've said, the FSR's resistance changes as more pressure is applied. When there is no pressure, the sensor looks like an infinite resistor (open circuit), as the pressure increases, the resistance goes down. This graph

indicates approximately the resistance of the sensor at different force measure ments.

It is important to notice that the graph isn't really *linear* (its a log/log graph) and that at especially low force measurements it quickly goes from infinite to100KΩ.

6 Wifi module (ESP8266)

High Durability

ESP8266EX is capable of functioning consistently in “industrial environments, due to its wide operating temperature range”. With highly-integrated on-chip features and minimal external discrete component count, the chip offers reliability, compactness and robustness.

Compactness

ESP8266EX is integrated with a “32-bit Tensilica processor, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules”. All of them are included in one small package, our ESP8266EX.

Power-Saving Architecture

Engineered for mobile devices, wearable electronics and IoT applications, “ESP8266EX achieves low power consumption with a combination of several proprietary technologies.” The power-saving architecture features three modes of operation: active mode, sleep mode and deep sleep mode. This allows battery-powered designs to run longer.

32-bit Tensilica Processor

The ESP8266EX microcontroller integrates a "Tensilica L106 32-bit RISC processor, which achieves extra-low power consumption and reaches a maximum clock speed of 160 MHz."The Real-Time Operating System (RTOS) and Wi-Fi stack allow about 80% of the processing power to be available for user application programming and development.

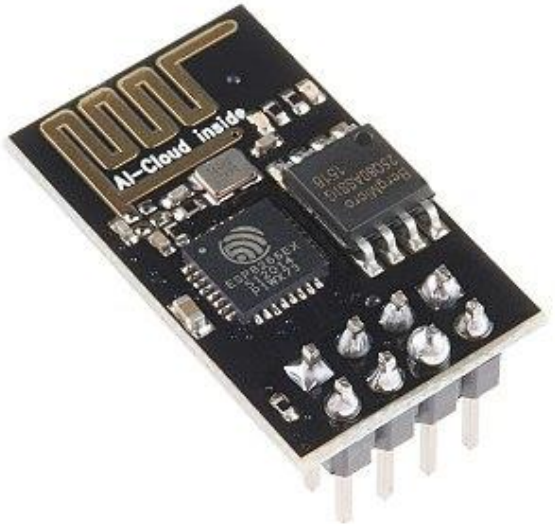

Fig.3.7 WIFI Module

3.3.2 Haptic Glove

The Haptic Glove is the controller for the robotic arm it comprises of ADXL-345 sensors placed on the fingertips of the glove to read the angle that the finger is making. The sensors have been placed in a round robin fashion to read the values from it and place them in the firebase database.

The haptic glove can be placed in any distance from the robotic arm but can still perform its duty to read and communicate with the arm in order to make it move.

The haptic glove user can also monitor the status of the robotic arm by using the monitor to see the temperatures of the object which is in contact with the arm, by which he could come to a decision to either continue holding it or let go.
The robotic arm also comes with a safety feature by which it will automatically let go of the object if it determines it to be damaging to the arm in any fashion. This feature could be turned off at the owner's own risk. We determined adxl-345 to be the best fit for our glove as it is capable of noticing changes in 3 of its axis, so say you use the abduction, the angle of the finger does not change but the position of it does. This kind of complex movements can be most effectively be recorded by using only the adxl-345 as it's a 3-d axis accelerometer.

3.3.1 Robotic Arm

Model Designing

3D models are created using Computer Aided Design (CAD) Softwares. Models are called as part files during the design phase. These part files are converted into .STL files by the CAD software which helps in 3D printing a model. 3D modelling is a process of creating virtual Three Dimensional object which is identical to a real life object. After modelling different parts of a project, the parts can be put into an assembly to perform animation and motion studies of the final assembly to test its movement capabilities of the assembly. After all the parts are finalized they are converted into .STL files convenient for 3D printing the part. STL file is a set of polygons connected together to form the model to be printed.

The basic ideology during the design of the robotic arm was to make sure that we create one which is capable of doing most of the movements that can be done by a human arm.

Few movements that we have tried and mimicked in the design include the abduction as in fig.1,folding of fingers with 2 servo motors as we are capable of folding just the pip and dip without moving the mcp joint of our hand. This required us to keep 2 servos per finger.

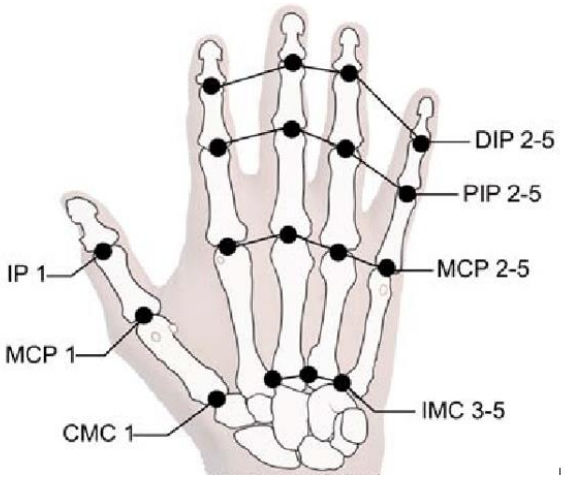

Fig.3.8 Palm Bone Joint Names

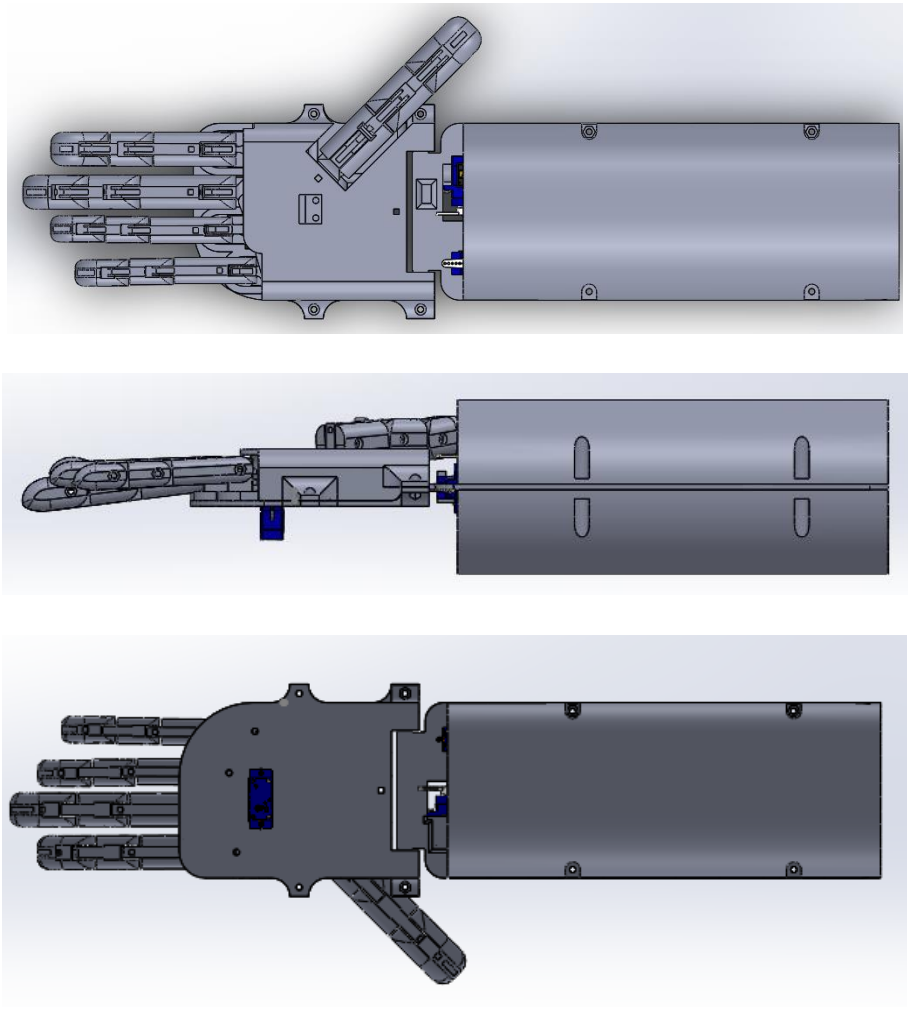

Fig.3.9 Final Robotic arm assembly

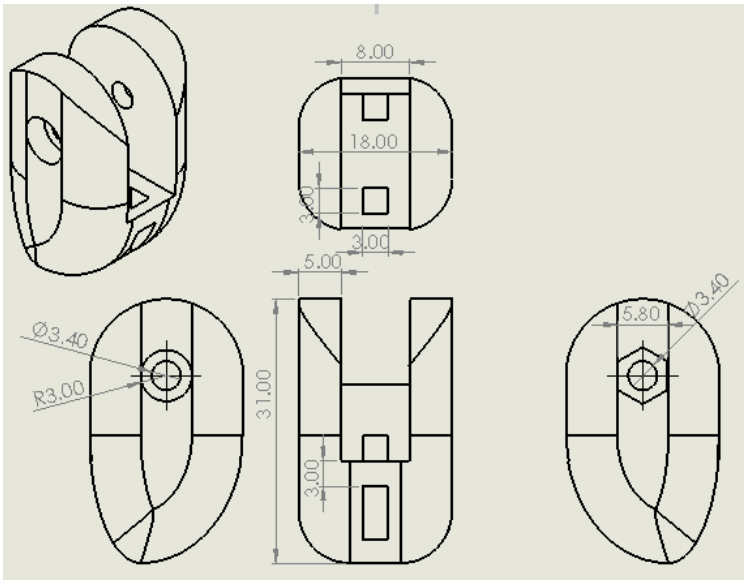

Fig.3.10 Index Finger above DIP joint

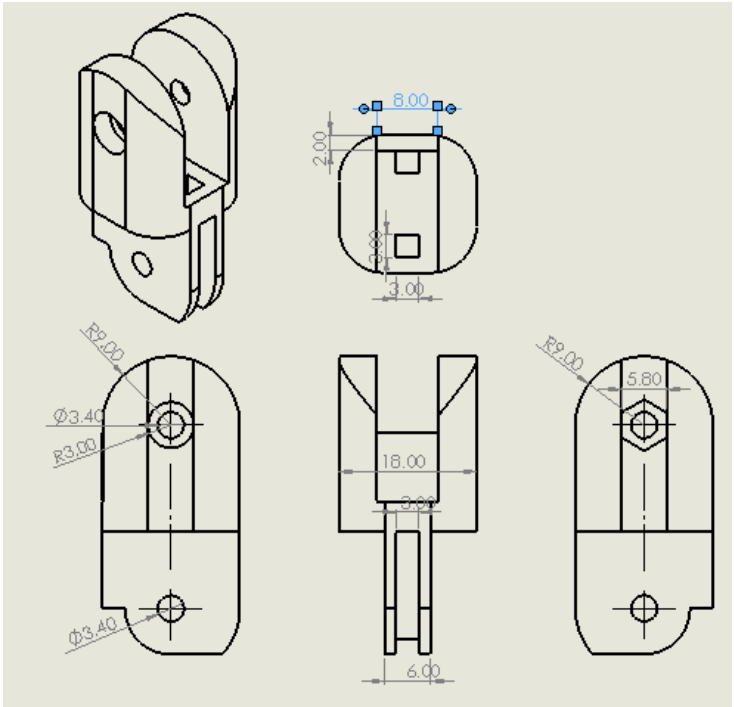

Fig.3.11 Index Finger between DIP and PIP joints

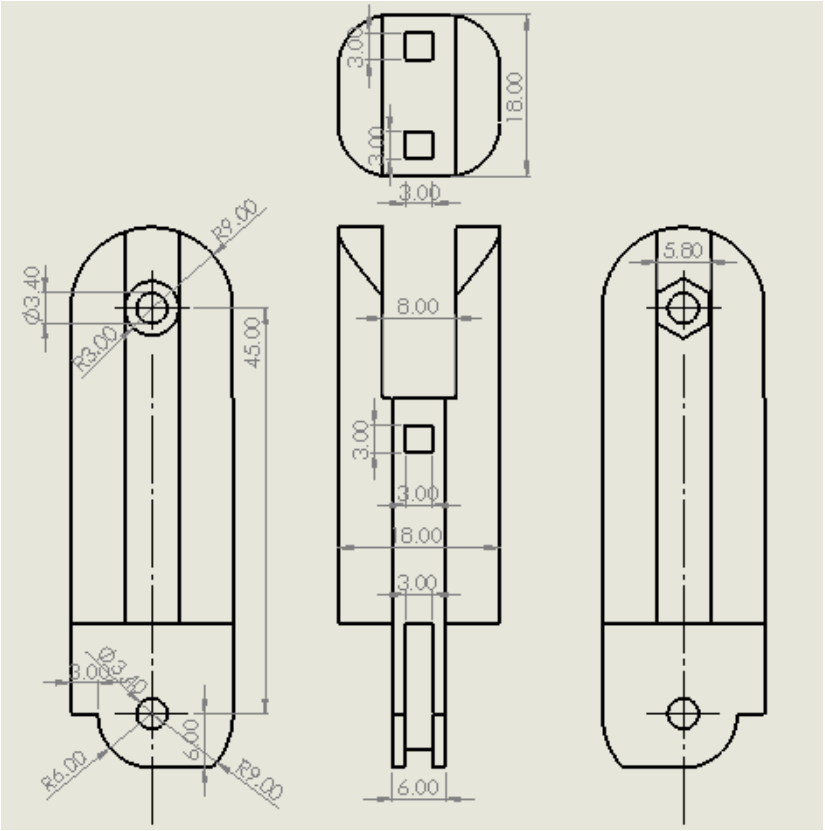

Fig.3.12 Index finger between PIP and MCP joints

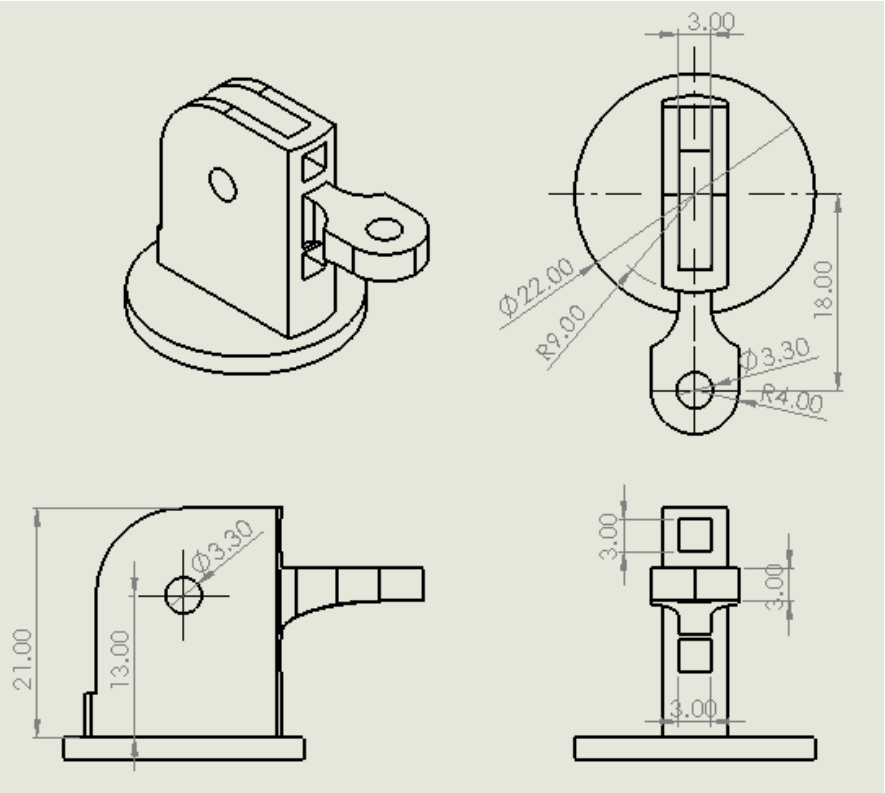

Fig.3.13 Index finger Knuckle

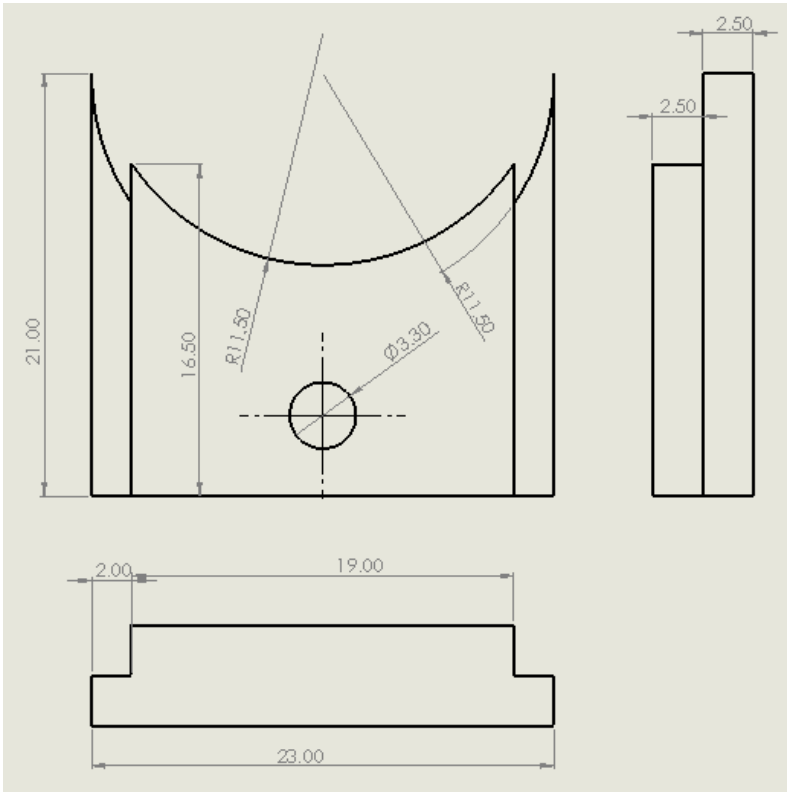

Fig.3.14 Index finger Knuckle joint stopper

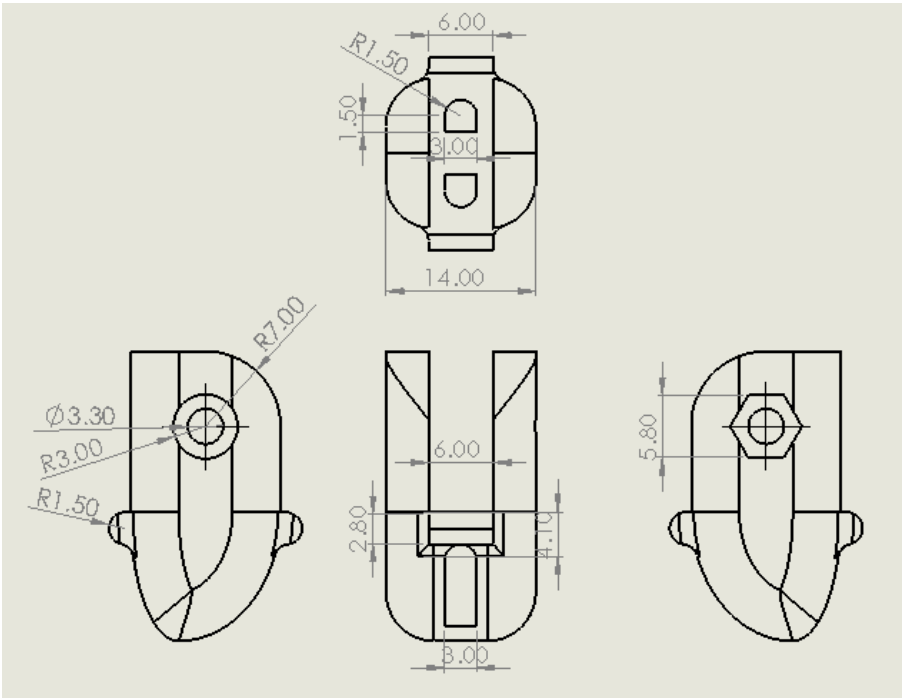

Fig.3.15 Little finger above DIP joint

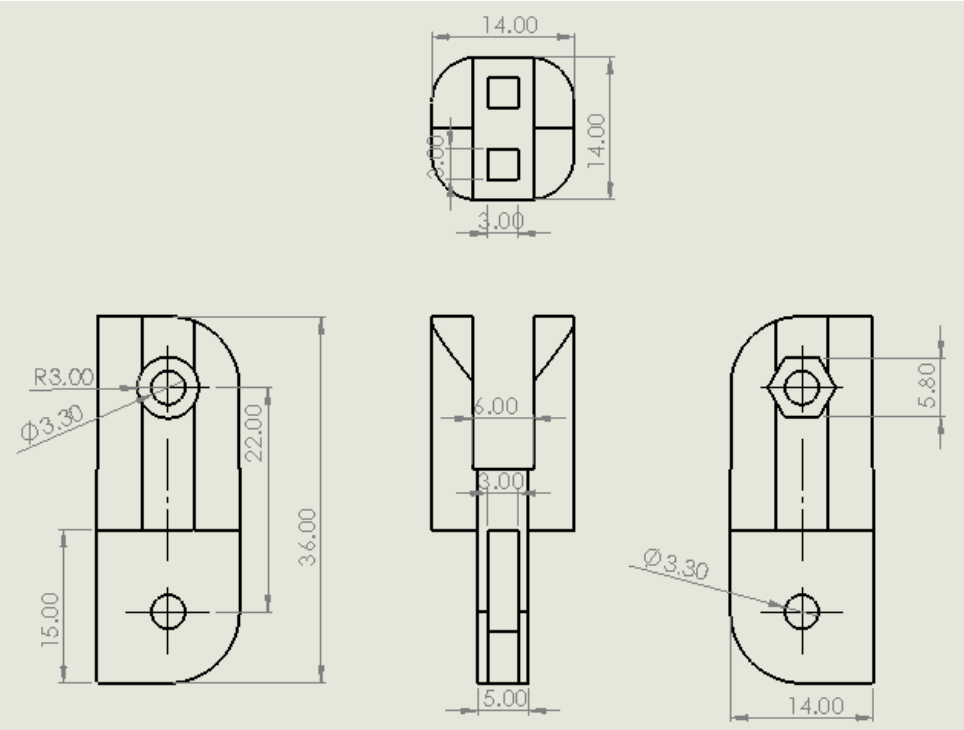

Fig.3.16 Little finger between DIP and PIP joints

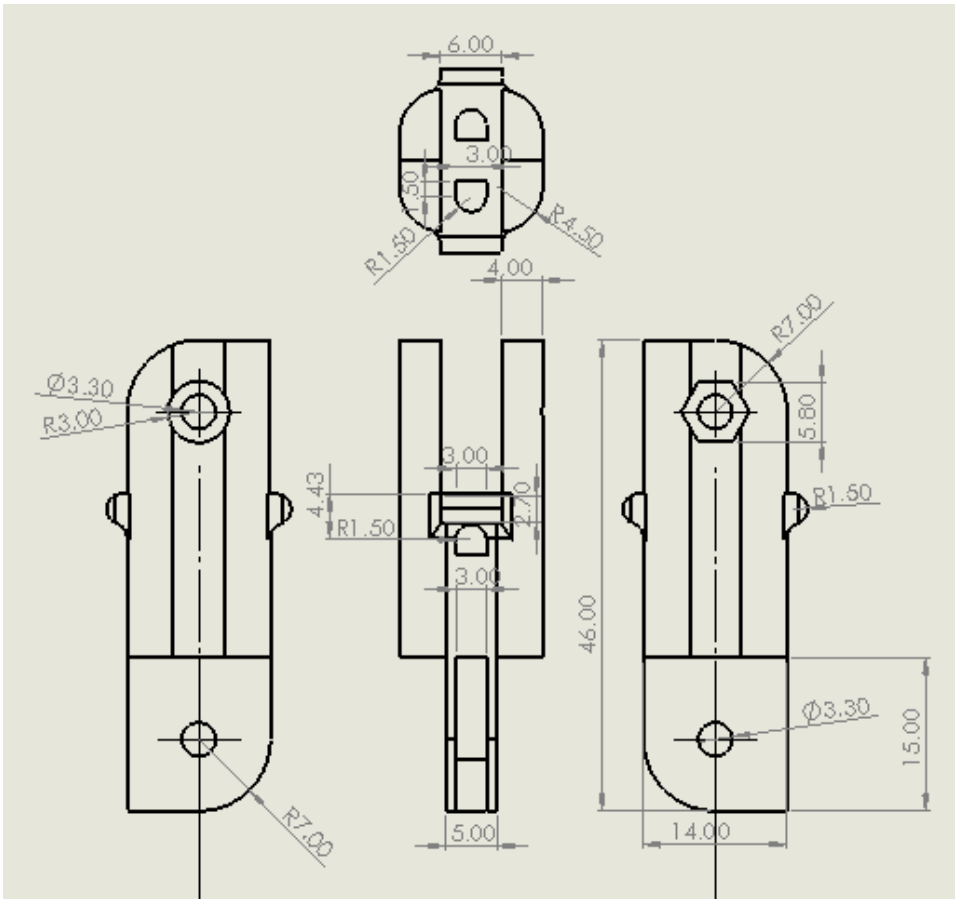

Fig.3.17 Little finger between PIP and MCP joints

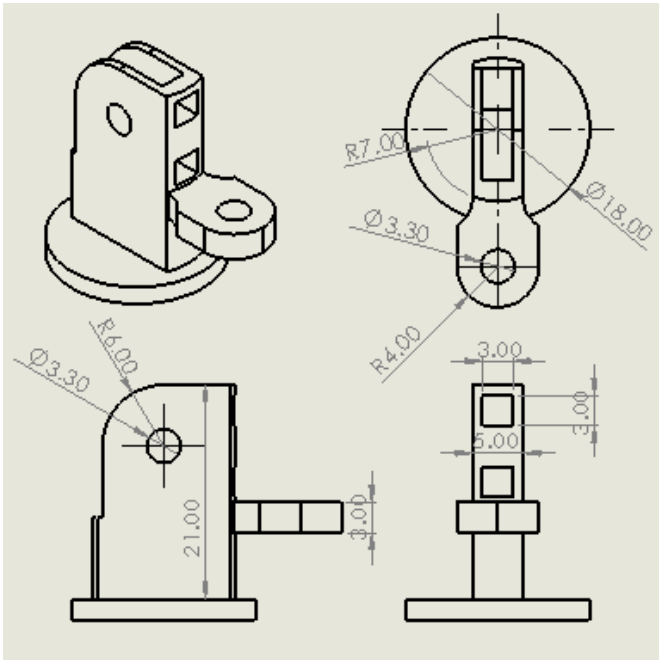

Fig.3.18 Little finger Knuckle

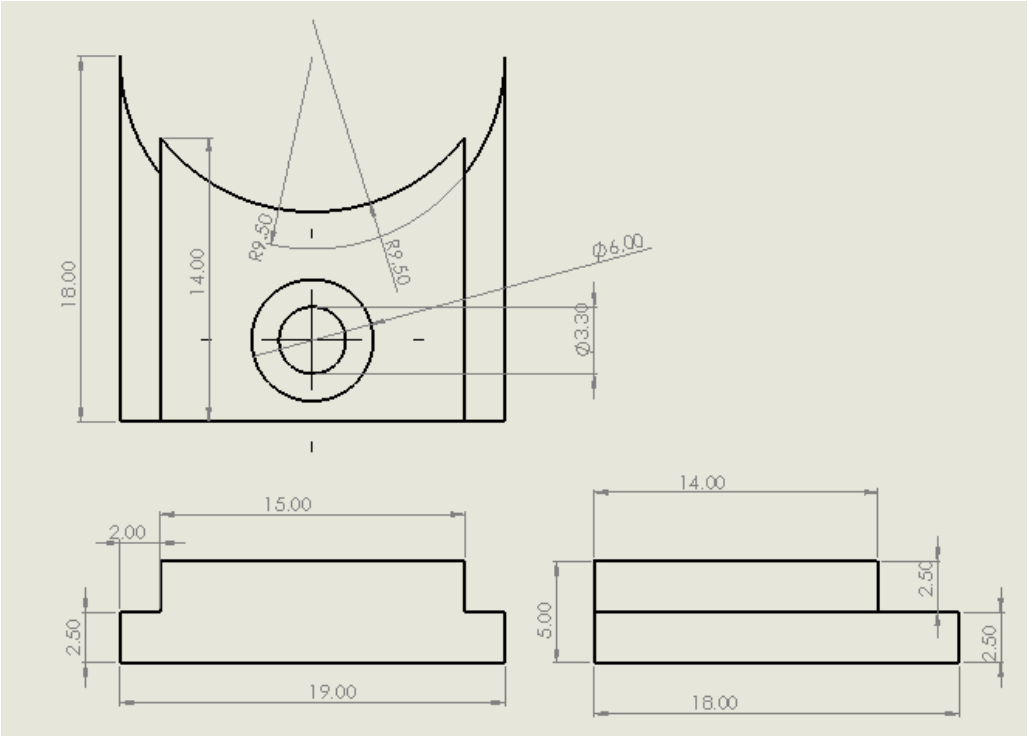

Fig.3.19 Little finger knuckle stopper

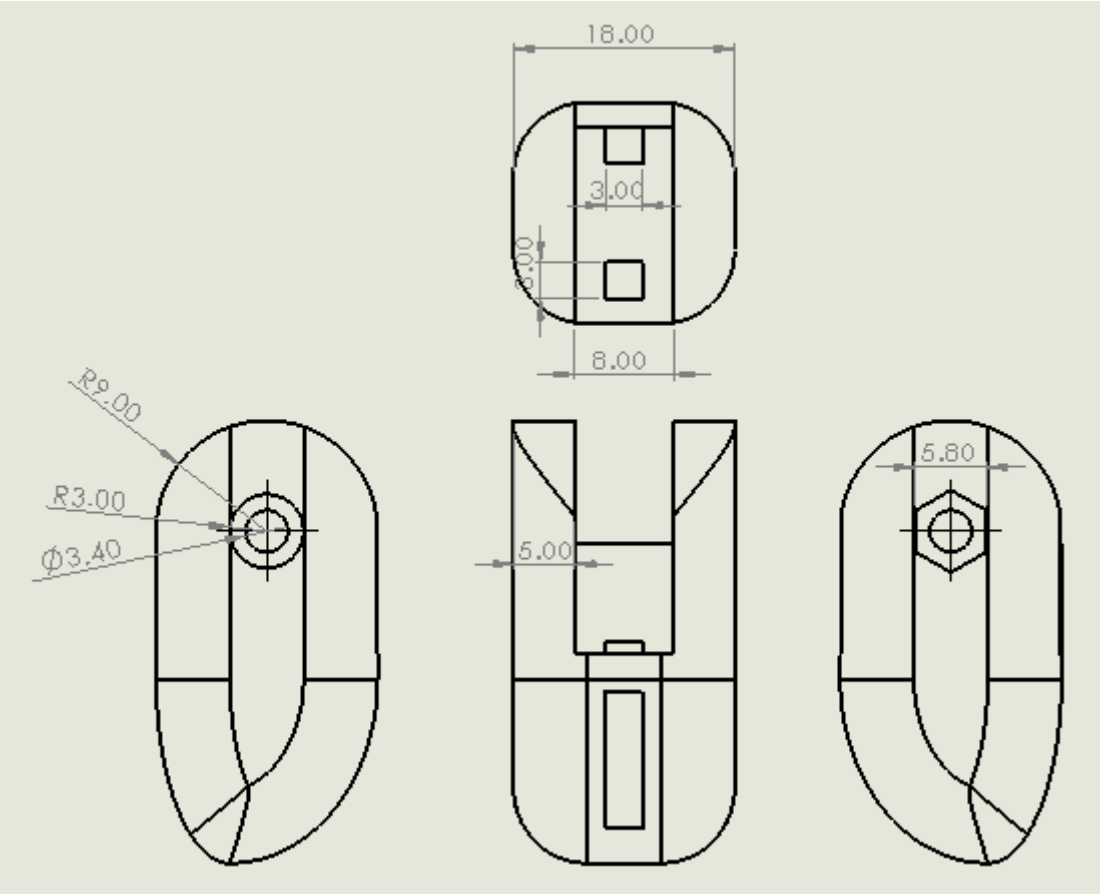

Fig.3.20 Middle finger above DIP joint

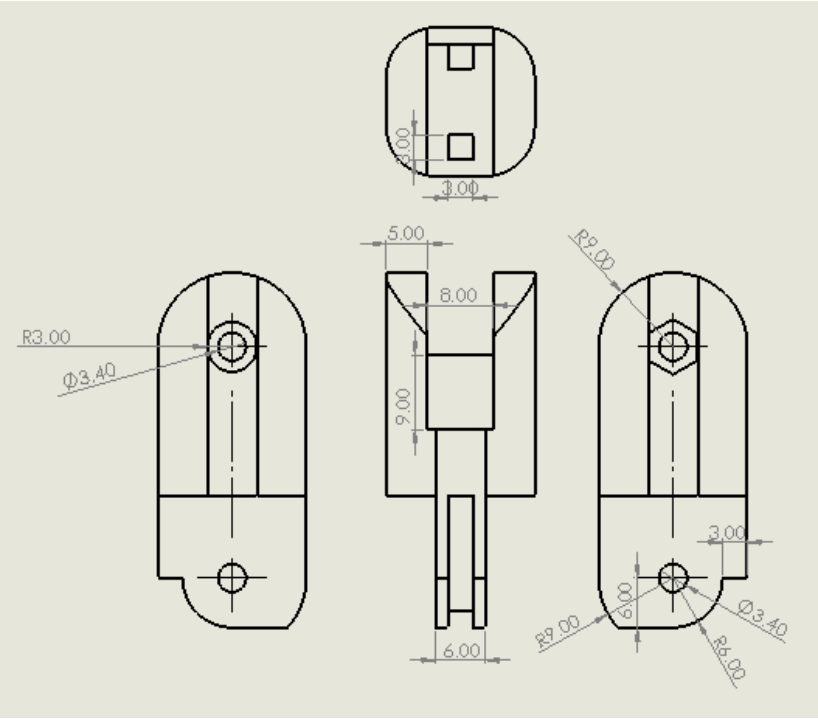

Fig.3.21 Middle finger between DIP and PIP joints

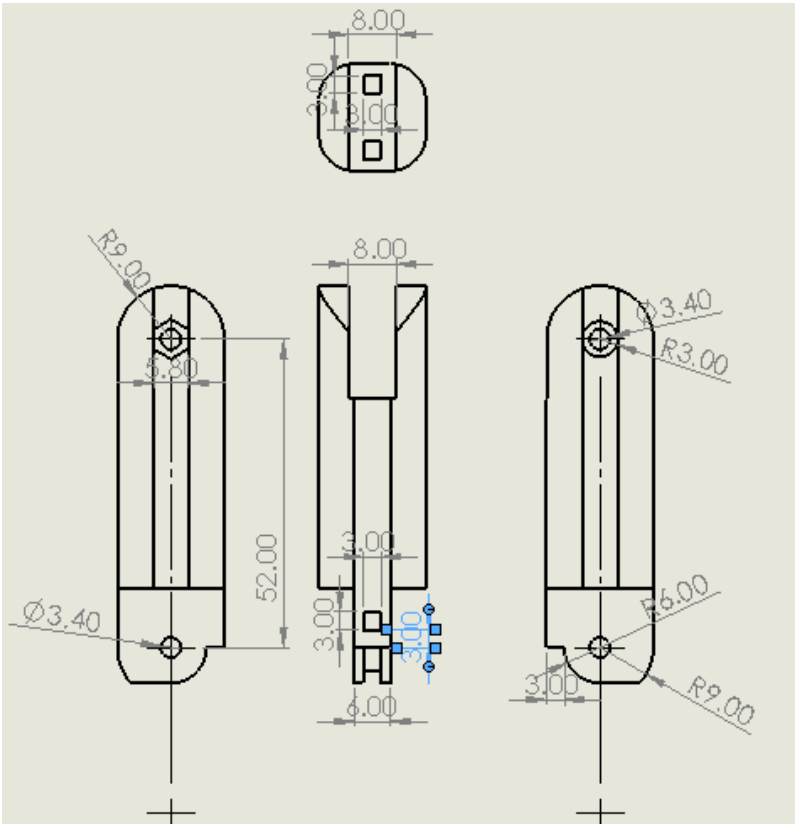

Fig.3.22 Middle finger between PIP and MCP joints

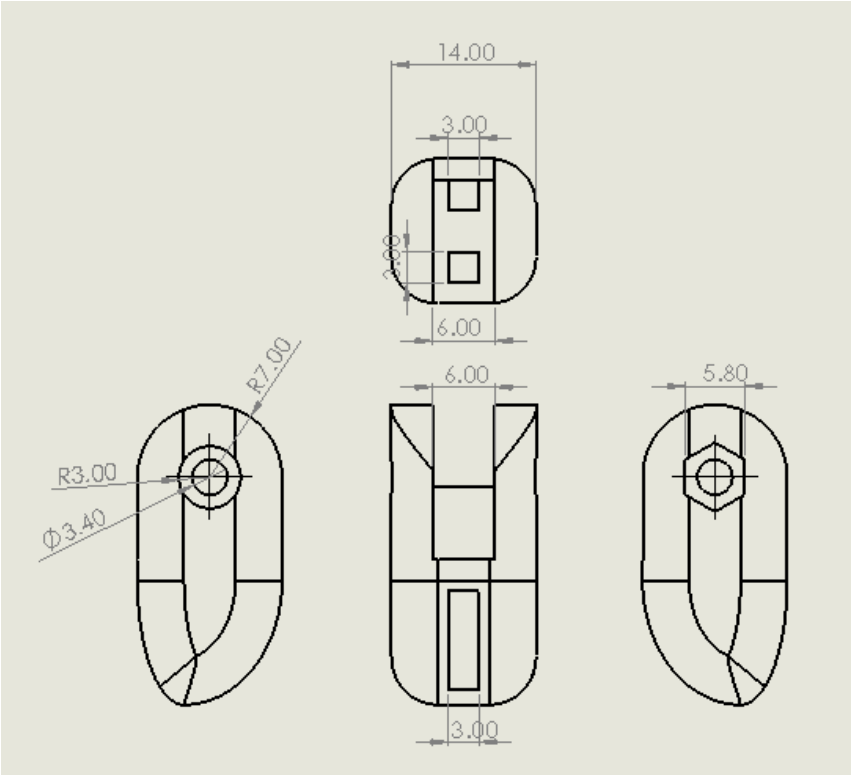

Fig.3.23 Ring finger above DIP joint

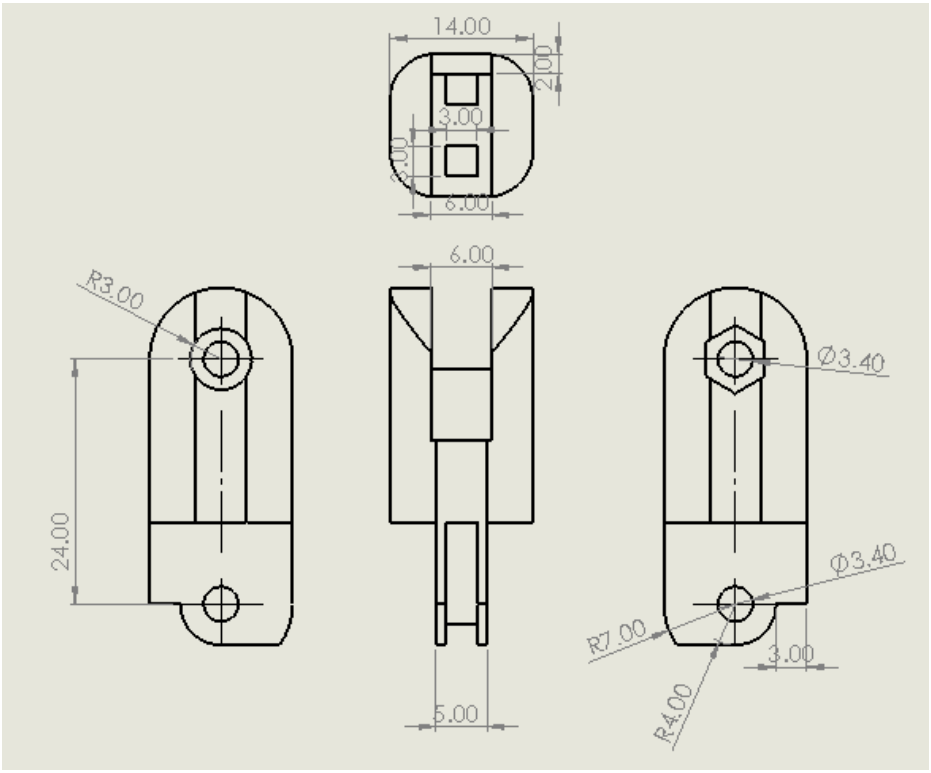

Fig.3.24 Ring finger between DIP and PIP joints

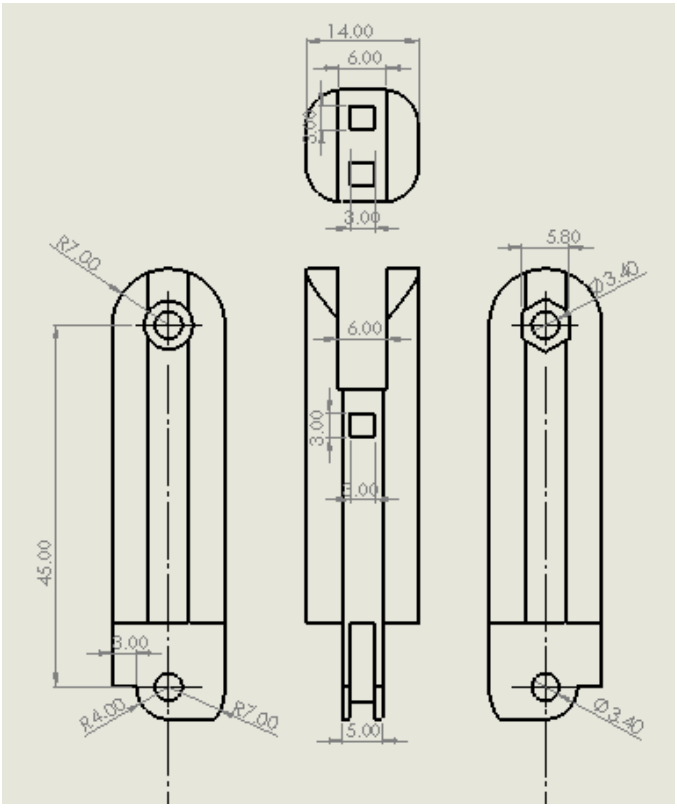

Fig.3.25 Ring finger PIP and MCP joints

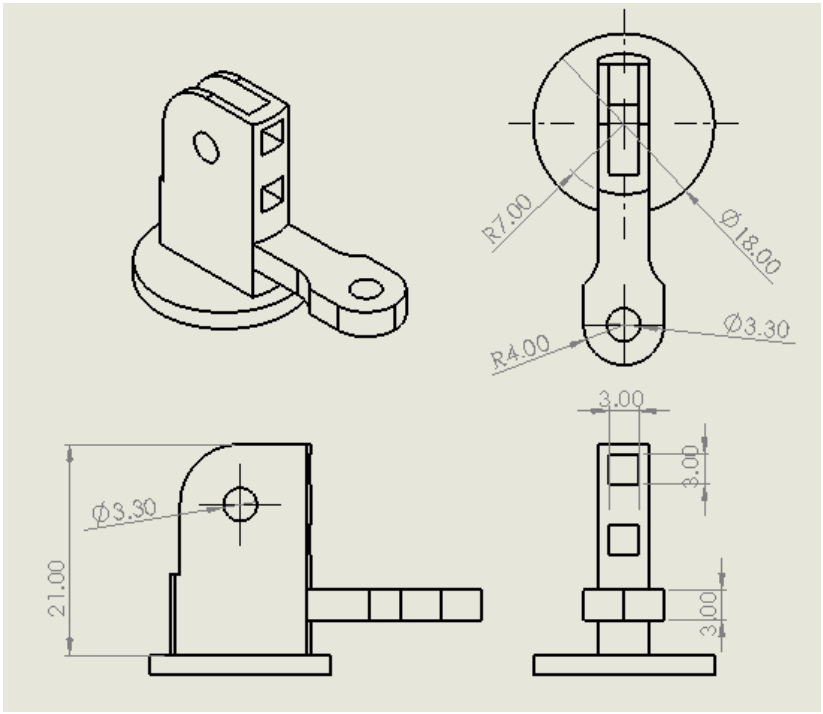

Fig.3.26 Ring finger knuckle

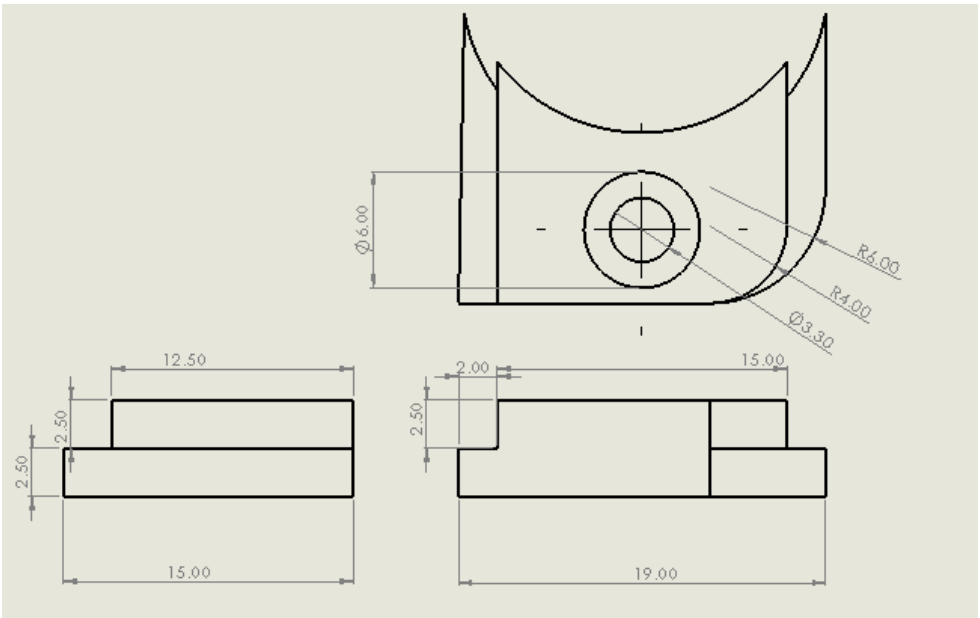

Fig.3.27 Ringer finger knuckle stopper

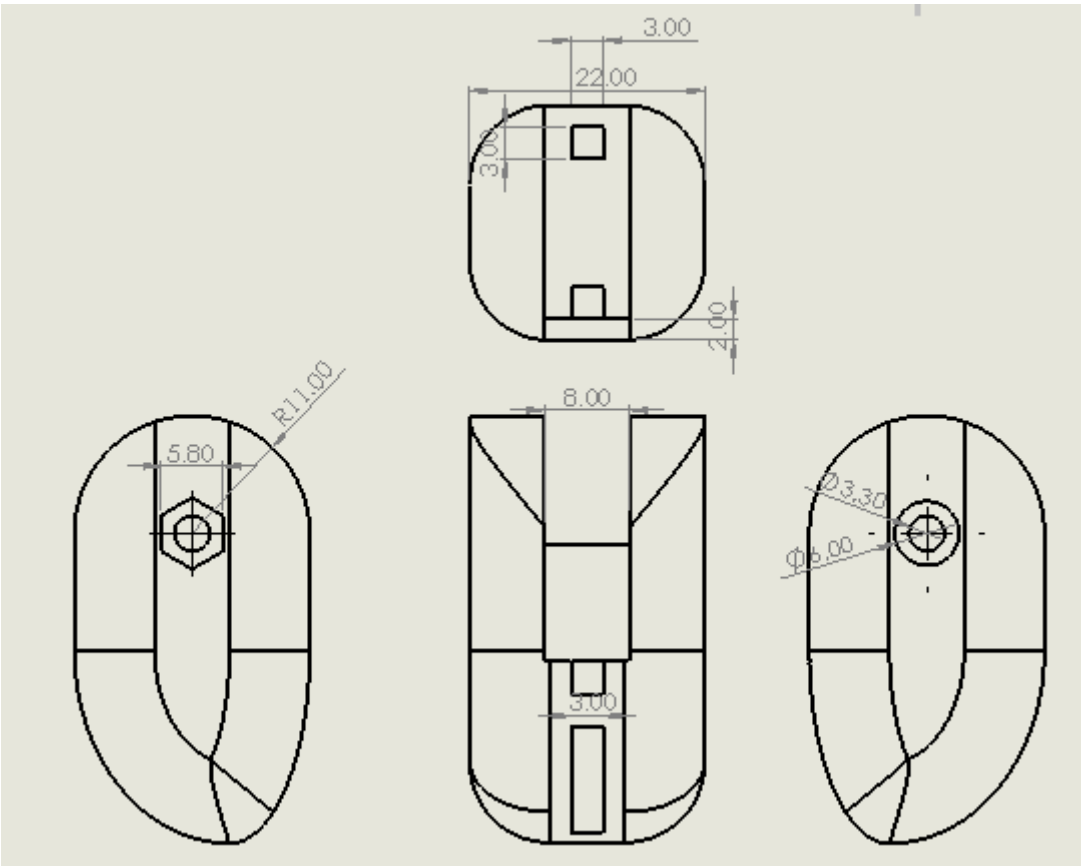

Fig.3.28 Thumb finger above DIP joint

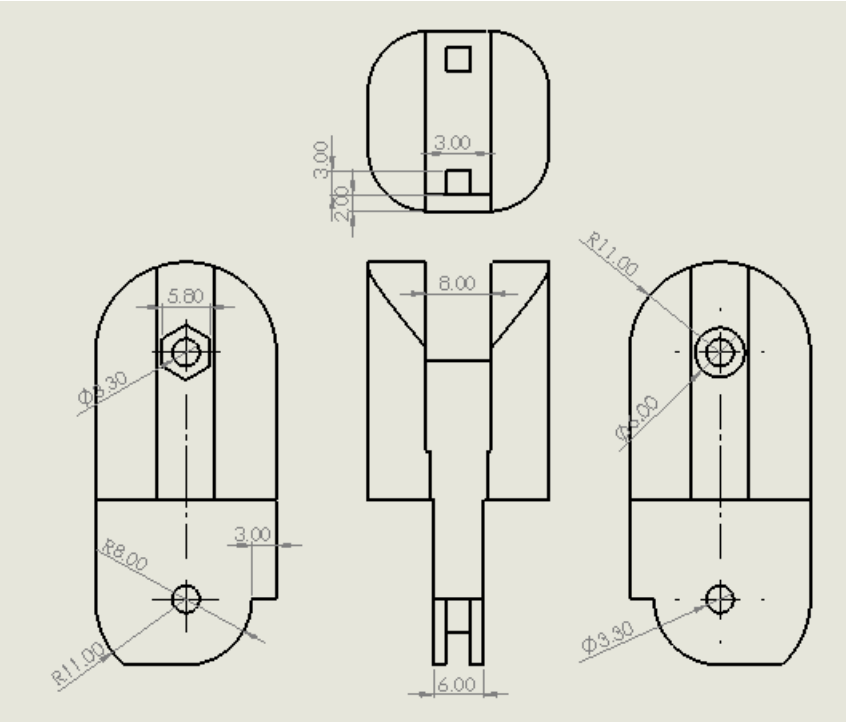

Fig.3.29 Thumb finger between DIP and PIP joints

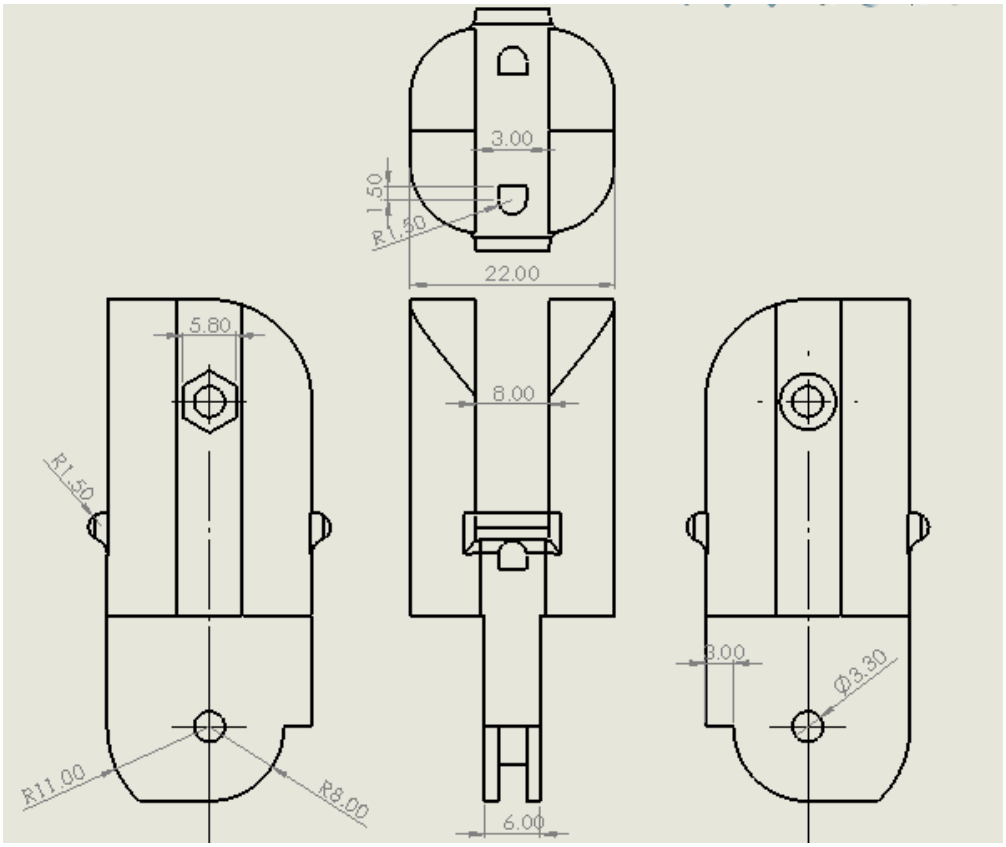

Fig.3.30 Thumb finger between PIP and MCP joint

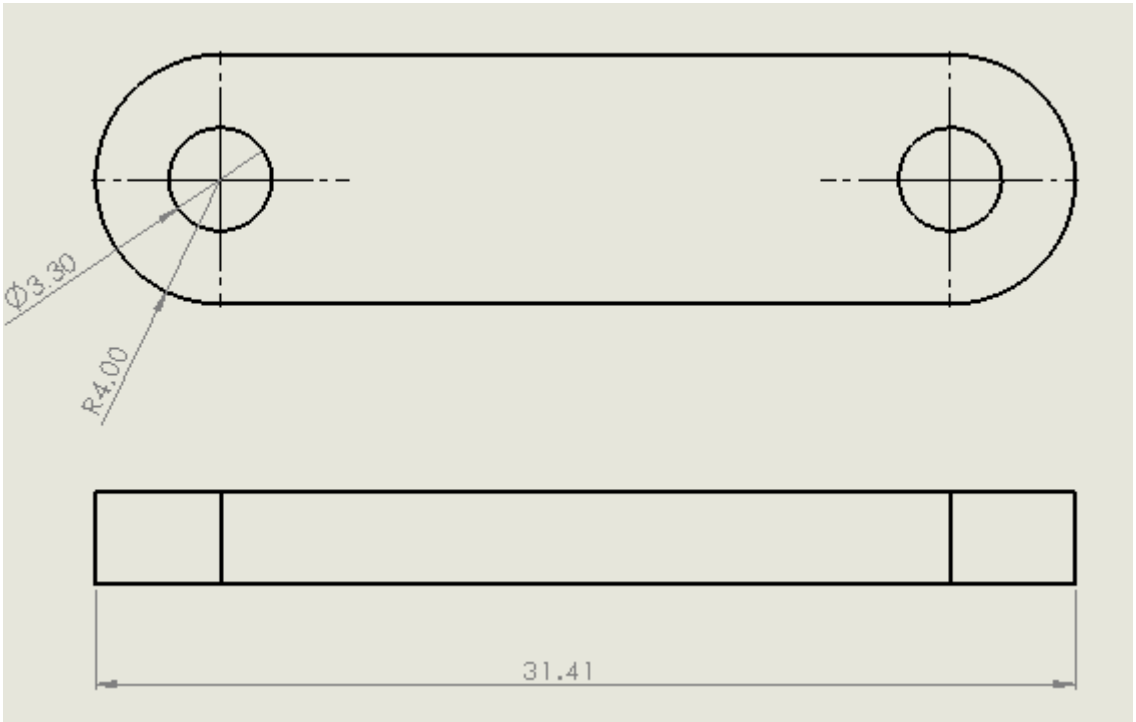

Fig.3.31 Ring finger to Little finger connection

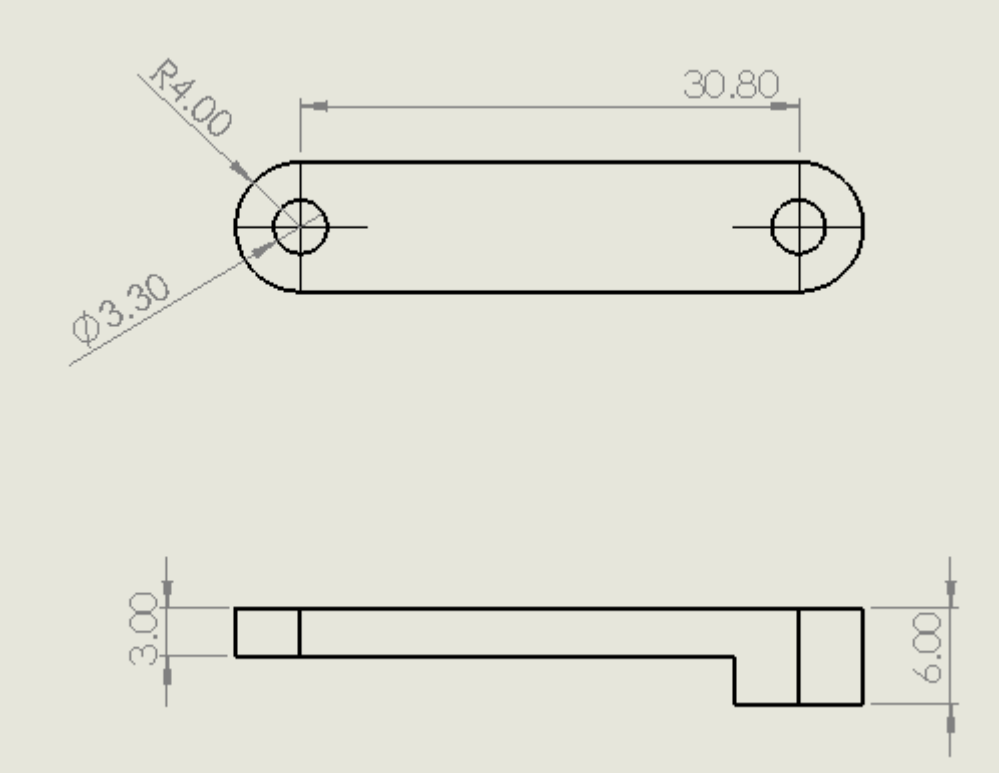

Fig.3.32 Servo to Index finger connection

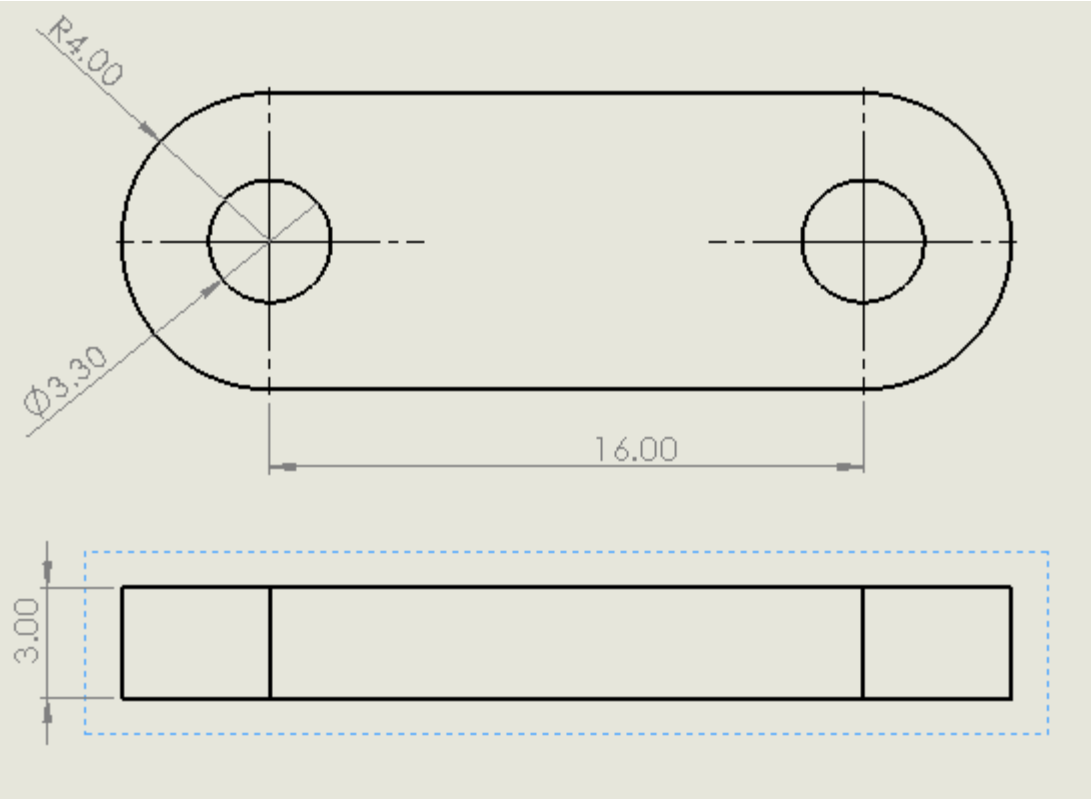

Fig.3.33 Servo to ring finger connection

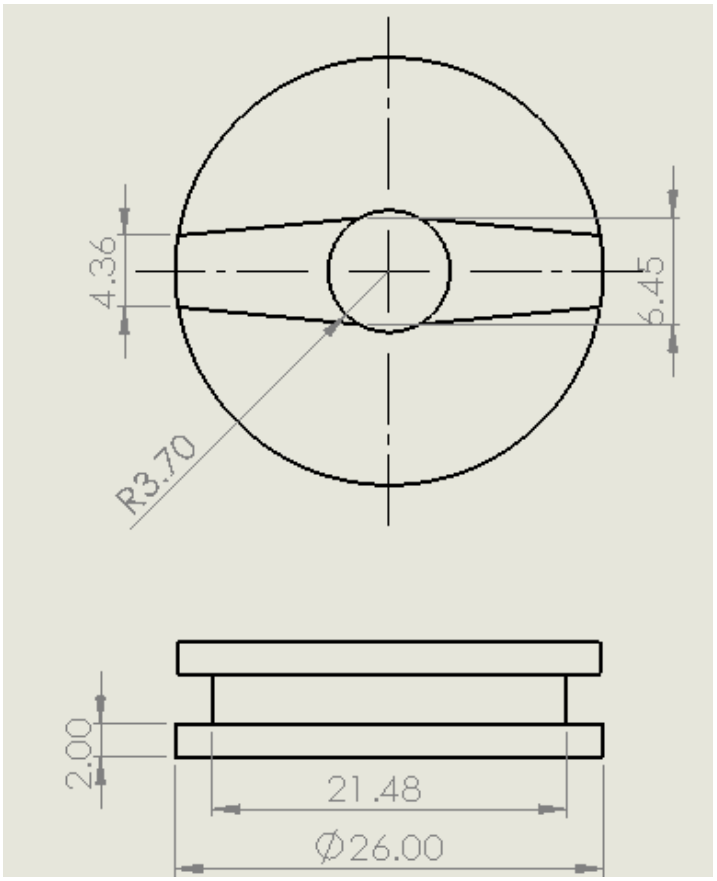

Fig.3.34 Servo cover for controlling fingers

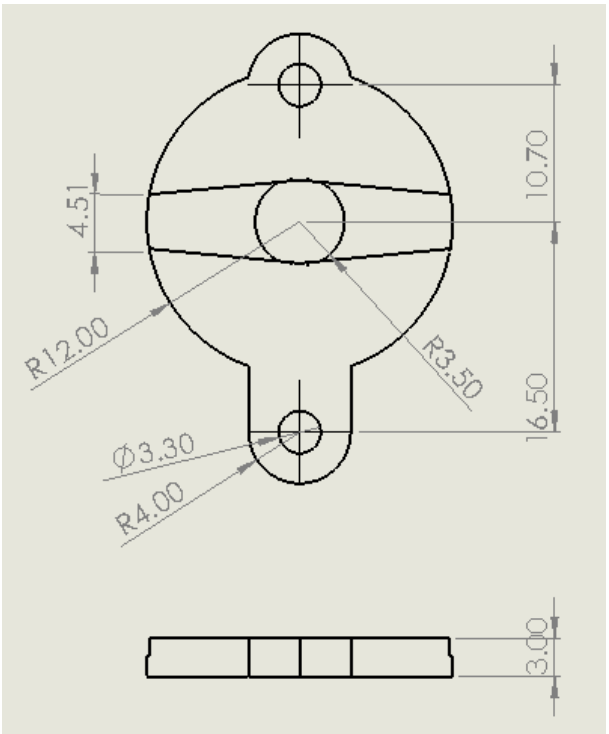

Fig.3.35 Servo cover for abduction mechanism

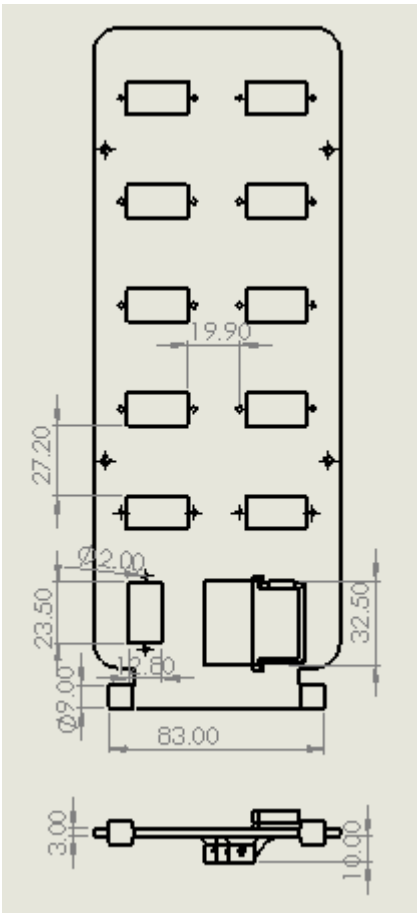

Fig.3.36 Servo mount for finger control

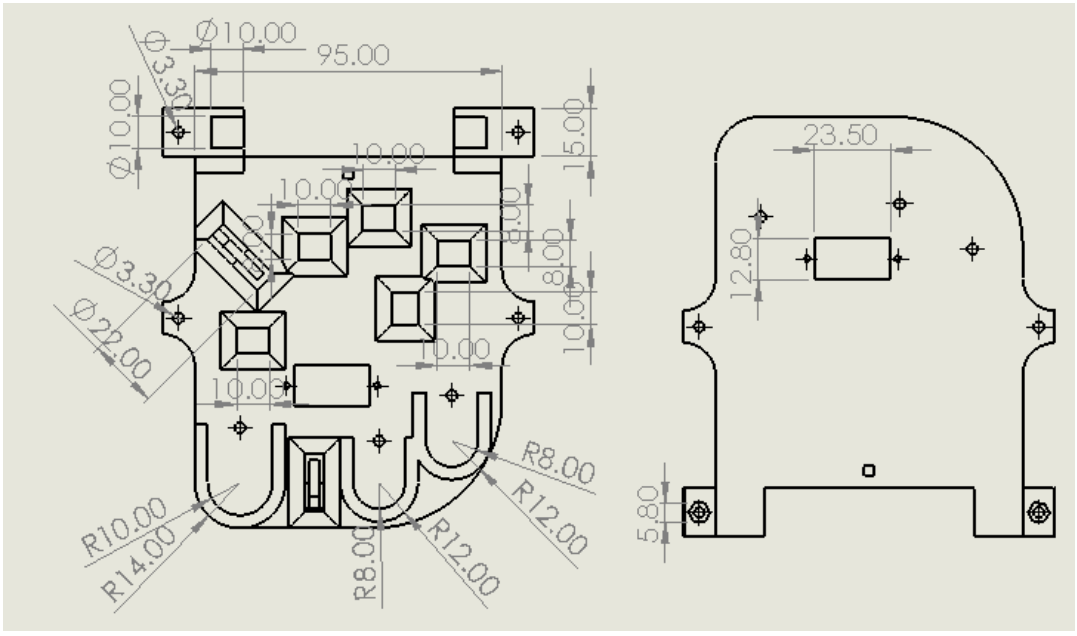

Fig.3.37 Palm back plate

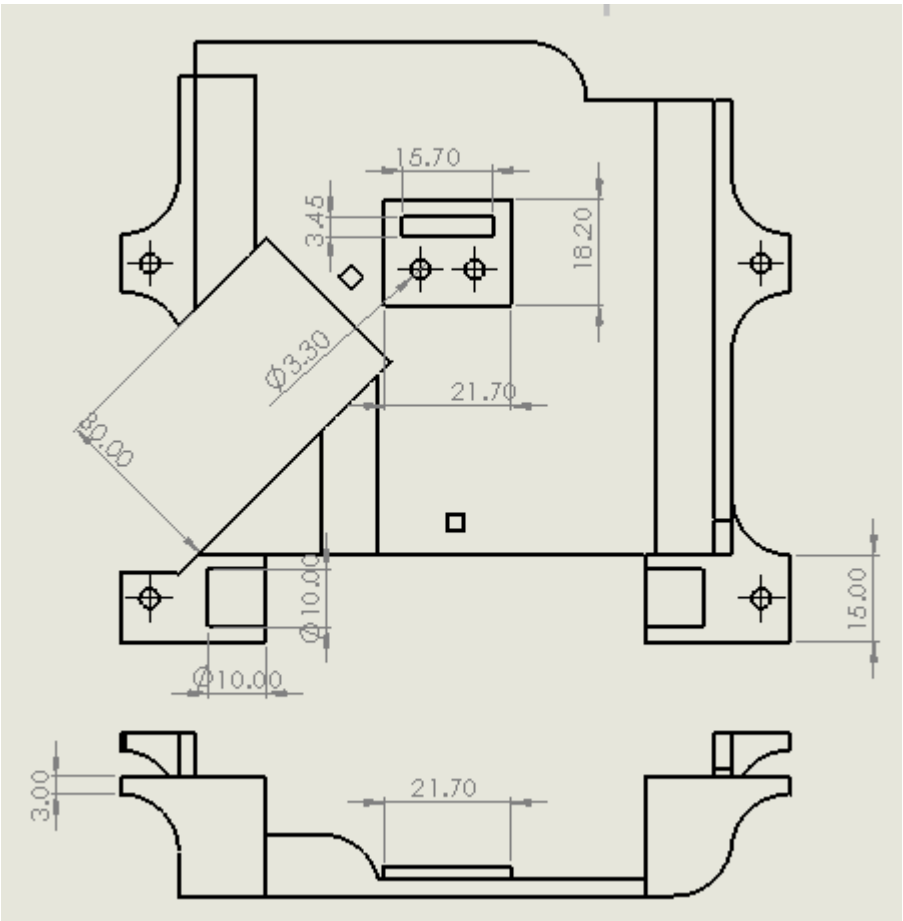

Fig.3.38 Palm front cover

3D printing

3D printing is a process of making an object by depositing layers of material in succession. First the .STL files created using CAD software is fed into a slicing software to make the model into layers. These layers when assembled on top of each other will give the object as a whole. The size of the layer depends on the extruder quality which can be varied during the slicing The material used for printing using the 3d printer was Pla. PLA (Polylactic Acid) is a biodegradable (under the correct conditions) thermoplastic derived from renewable resources such as corn starch or sugarcane. It is one of the most popular bioplastics, used for many applications ranging from plastic cups to medical implants.

All the parts were printed with various fillings and different speeds to find a reasonable balance of cost vs pressure exerted on it.

The printer type that we have used for printing is of type FDM. Fdm printers work on the principle of them melting the thermoplastic filament and extruding this melted substance layer by layer in the 3-d shape that we desire. The extruder is fixed to a body. This entire body moves in 3-dimentions to print. The position in which you would like to print the object also plays an important role as the amount of substance required to print the same object, the time it requires greatly vary. The finishing that you get also varies in each printing position.

The amount of substance varies as depending on its orientation the printer decides how much and what shape the foundation must be in order to print the final desired structure without it breaking apart, or it being able to reach the position to let filament out.

The extruder of the 3-d printer is maintained at a high 200°C, whereas the bed onto which the object is printed is at 60°C this is to prevent the base layer from sticking onto the bed but also attach itself slightly in order to avoid unnecessary movements to print the object successfully.

3.4 Software architecture

3.4.1 Code

Esp 8266 writing code for the database:

```
#include <ESP8266WiFi.h>

#include <FirebaseArduino.h>
```

```
// Set these to run example.
#define FIREBASE_HOST "roboticarmcontrol.firebaseio.com"
#define FIREBASE_AUTH "HDyPp3e2SGe3SXmutrI9gqYQ5FGY9MrvhaX960bG"
#define WIFI_SSID "Anirudh"
#define WIFI_PASSWORD "whysayno"


 void setup() {
Serial.begin(9600);
 // connect to wifi.
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
 Serial.print("connecting");
 while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
 }
  Serial.println();
 Serial.print("connected: ");
 Serial.println(WiFi.localIP());
 Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH  );
}
//initialize values so as to crate the variables onto the database.
float a=20.261;
```

```
float b=3.310;

float c=40.213;

float d=50.1232;

float e=60.21346;


int n = 0;

void loop() {

 // update value

 Firebase.setFloat("Thumb",a);

 Firebase.setFloat("Index", b);

 Firebase.setFloat("Middle", c);

 Firebase.setFloat("Ring", d);

 Firebase.setFloat("Pinky", e);

 //   handle error

 if (Firebase.failed()) {

    Serial.print("setting /number failed:");

    Serial.println(Firebase.error());

    return;

 }

 Serial.println("pushed: /logs/");

 delay(1000  );

}
```

Esp 8266 receiver code from database:

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "roboticarmcontrol.firebaseio.com"
#define FIREBASE_AUTH "HDyPp3e2SGe3SXmutrI9gqYQ5FGY9MrvhaX960bG"
#define WIFI_SSID "Anirudh"
#define WIFI_PASSWORD "whysayno"
 void setup() {
Serial.begin(115200);
 // connect to wifi.
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
 Serial.print("connecting");
 while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
 }
 Serial.println();
 Serial.print("connected: ");
 Serial.println(WiFi.localIP());
```

```
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
void loop() {
 // get value
  Serial.print("Thumb: ");
  Serial.println(Firebase.getFloat("Thumb"));
  Serial.print("Index: ");
  Serial.println(Firebase.getFloat("Index"));
  Serial.print("Middle: ");
  Serial.println(Firebase.getFloat("Middle"));
  Serial.print("Ring: ");
  Serial.println(Firebase.getFloat("Ring"));
  Serial.print("Pinky: ");
  Serial.println(Firebase.getFloat("Pinky"));
  delay(1000);
}
```

Glove's Accelerometer code-

```
 #include <SparkFun_ADXL345_robot.h>
ADXL345 adxl_0 = ADXL345(10);
ADXL345 adxl_1 = ADXL345(9);
ADXL345 adxl_2 = ADXL345(8);
```

```
ADXL345 adxl_3 = ADXL345(7);

ADXL345 adxl_4 = ADXL345(6);

//ADXL345 adxl = ADXL345();           // USE FOR I2C COMMUNICATION

void setup(){

  Serial.begin(9600);

  Serial.println("SparkFun ADXL345 Accelerometer ");

  Serial.println();

 }

void loop(){

  ADXL_0_Data();

  delay(100);

  ADXL_1_Data();

  delay(100);

  ADXL_2_Data();

  delay(100);

  ADXL_3_Data();

  delay(100);

  ADXL_4_Data();

  delay(100);

}

void ADXL_0_setup(){

  adxl_0.powerOn();                 //Wake up the Senosr

    adxl_0.setRangeSetting(16);
```

```
// Accepted values gravitational ranges  are 2g, 4g, 8g or 16g
// Higher 'g' Values are  = Wider Measurement Range
// Lower 'g' Values = Greater Sensitivity
  adxl_0.setSpiBit(0);              //SPI configuration
adxl_0.setActivityXYZ(1, 0, 0);
adxl_0.setActivityThreshold(75);
adxl_0.setInactivityXYZ(1, 0, 0);
adxl_0.setInactivityThreshold(75);
adxl_0.setTimeInactivity(10);
adxl_0.setTapDetectionOnXYZ(0, 0, 1);
adxl_0.setTapThreshold(50);
adxl_0.setTapDuration(15);
adxl_0.setDoubleTapLatency(80);
adxl_0.setDoubleTapWindow(200);
adxl_0.setFreeFallThreshold(7);
adxl_0.setFreeFallDuration(30);      //free fall ranges
}
 void ADXL_1_setup(){
adxl_1.powerOn();
adxl_1.setRangeSetting(16);
adxl_1.setSpiBit(0);
adxl_1.setActivityXYZ(1, 0, 0);
adxl_1.setActivityThreshold(75);
```

```
adxl_1.setInactivityXYZ(1, 0, 0);

adxl_1.setInactivityThreshold(75);

adxl_1.setTimeInactivity(10);

adxl_1.setTapDetectionOnXYZ(0, 0, 1);

adxl_1.setTapThreshold(50);

adxl_1.setTapDuration(15);

adxl_1.setDoubleTapLatency(80);

adxl_1.setDoubleTapWindow(200);

adxl_1.setFreeFallThreshold(7);

adxl_1.setFreeFallDuration(30);

}


void ADXL_2_setup(){

adxl_2.powerOn();

adxl_2.setRangeSetting(16);

adxl_2.setSpiBit(0);

 adxl_2.setActivityXYZ(1, 0, 0);

adxl_2.setActivityThreshold(75);

adxl_2.setInactivityXYZ(1, 0, 0);

adxl_2.setInactivityThreshold(75);

adxl_2.setTimeInactivity(10);

adxl_2.setTapDetectionOnXYZ(0, 0, 1);

adxl_2.setTapThreshold(50);
```

```
adxl_2.setTapDuration(15);

adxl_2.setDoubleTapLatency(80);

adxl_2.setDoubleTapWindow(200);

adxl_2.setFreeFallThreshold(7);

adxl_2.setFreeFallDuration(30);

}


void ADXL_3_setup(){

adxl_3.powerOn();

 adxl_3.setRangeSetting(16);

adxl_3.setSpiBit(0);

adxl_3.setActivityXYZ(1, 0, 0);

adxl_3.setActivityThreshold(75);

adxl_3.setInactivityXYZ(1, 0, 0);

adxl_3.setInactivityThreshold(75);

adxl_3.setTimeInactivity(10);

adxl_3.setTapDetectionOnXYZ(0, 0, 1);

adxl_3.setTapThreshold(50);

adxl_3.setTapDuration(15);

adxl_3.setDoubleTapLatency(80);

adxl_3.setDoubleTapWindow(200);

adxl_3.setFreeFallThreshold(7);

adxl_3.setFreeFallDuration(30);
```

```
}


void ADXL_4_setup(){
adxl_4.powerOn();
 adxl_4.setRangeSetting(16);
adxl_4.setSpiBit(0);
adxl_4.setActivityXYZ(1, 0, 0);
adxl_4.setActivityThreshold(75);
adxl_4.setInactivityXYZ(1, 0, 0);
adxl_4.setInactivityThreshold(75);
adxl_4.setTimeInactivity(10);
adxl_4.setTapDetectionOnXYZ(0, 0, 1);
adxl_4.setTapThreshold(50);
adxl_4.setTapDuration(15);
adxl_4.setDoubleTapLatency(80);
adxl_4.setDoubleTapWindow(200);
adxl_4.setFreeFallThreshold(7);
adxl_4.setFreeFallDuration(30);
}


void ADXL_0_Data(){
 digitalWrite(6, LOW);
 digitalWrite(7, LOW);
```

```
 digitalWrite(8, LOW);
 digitalWrite(9, LOW);
 digitalWrite(10, HIGH);
 delay(1000);
 ADXL_0_setup();
delay(1000);
int x0,y0,z0;
adxl_0.readAccel(&x0, &y0, &z0);
Serial.println("ADXL_0_Data\n");
Serial.print(x0);
Serial.print(", ");
Serial.print(y0);
Serial.print(", ");
Serial.println(z0);
Serial.print("Pitch Angle is : ");
Serial.print(acos(z0/(sqrt(pow(x0,2)+pow(y0,2)+pow(z0,2))))*(180/PI));
Serial.print("  ");
Serial.println("in degree ");
 digitalWrite(10,LOW);
 }

 void ADXL_1_Data(){
digitalWrite(6, LOW);
```

```
digitalWrite(7, LOW);

digitalWrite(8, LOW);

digitalWrite(9, HIGH);

digitalWrite(10,LOW);

delay(1000);

ADXL_1_setup();

delay(1000);



int x1,y1,z1;

adxl_1.readAccel(&x1, &y1, &z1);

Serial.println("ADXL_1_Data\n");

Serial.print(x1);

Serial.print(", ");

Serial.print(y1);

Serial.print(", ");

Serial.println(z1);

Serial.print("Pitch Angle is : ");

Serial.print(acos(z1/(sqrt(pow(x1,2)+pow(y1,2)+pow(z1,2))))*(180/PI));

Serial.print("  ");

Serial.println("in degree ");

digitalWrite(9,LOW);
```

```
}


void ADXL_2_Data(){
  digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
digitalWrite(9, LOW);
digitalWrite(10,LOW);
delay(1000);
 ADXL_2_setup();
 delay(1000);


  // Accelerometer Readings
int x2,y2,z2;
adxl_2.readAccel(&x2, &y2, &z2);
 // Read the accelerometer values and store them in variables declared above x,y,z
Serial.println("ADXL_2_Data\n");
Serial.print("Pitch Angle is : ");
Serial.print(acos(z2/(sqrt(pow(x2,2)+pow(y2,2)+pow(z2,2))))*(180/PI));
Serial.print("  ");
Serial.println("in degree ");
  }
 void ADXL_3_Data(){
```

```
 digitalWrite(6, LOW);
 digitalWrite(7, HIGH);
 digitalWrite(8, LOW);
 digitalWrite(9, LOW);
 digitalWrite(10,LOW);
 delay(1000);
 ADXL_3_setup();
 delay(1000);
int x3,y3,z3;
adxl_3.readAccel(&x3, &y3, &z3);
Serial.println("ADXL_3_Data\n");
 Serial.print("Pitch Angle is : ");
Serial.print(acos(z3/(sqrt(pow(x3,2)+pow(y3,2)+pow(z3,2))))*(180/PI));
Serial.print("  ");
Serial.println("in degree ");
 }
 void ADXL_4_Data(){
 digitalWrite(6, HIGH);
 digitalWrite(7, LOW);
 digitalWrite(8, LOW);
 digitalWrite(9, LOW);
 digitalWrite(10,LOW);
 ADXL_4_setup();
```

```
  int x4,y4,z4;
 adxl_4.readAccel(&x4, &y4, &z4);
 Serial.println("ADXL_4_Data\n");
 Serial.print("Pitch Angle is : ");
 Serial.print(acos(z4/(sqrt(pow(x4,2)+pow(y4,2)+pow(z4,2))))*(180/PI));
  Serial.print("  ");Serial.println("in degree ");
  }
```

Code for Testing of the arm:

```
 #include<Servo.h>
Servo index_dip,index_mcp;
int index_dip_val=135;
int index_mcp_val=180;
//                dip  mcp
//only dip,pip close  135  180
//fully extended     0    180
//fully closed       180  100
//only mcp closed    30   100
void setup()
{
 index_dip.attach(10);
 index_mcp.attach(9);
}
void loop()
```

```
{
  index_dip_run(0);
  index_mcp_run(180);
  delay(2000);
  index_dip_run(30);
  index_mcp_run(100);
  delay(2000);
  index_dip_run(180);
  index_mcp_run(100);
  delay(2000);
  index_dip_run(135);
  index_mcp_run(180);
  delay(2000);
}
void index_dip_run(int val)
{
  index_dip.write(val);
}
void index_mcp_run(int val)
{
  index_mcp.write(val);
}
```

CHAPTER 4: RESULTS AND DISCUSSION

The final outcome of the project is an easy-to-use, secure Robotic Arm which can be controlled via a haptic glove. The results can be discussed as the following:

Version 1 results:

The Version 1 consisted of many problems which included

- Delay in communication between the glove and the arm.
- Material used for printing was found to be of wrong type and settings used were wrong, making the hand brittle.
- The joint to move the entire palm was not working as expected.
- The servo bed was designed wrong due to which the servos could not sit stably.
- The Abduction mechanism was not working as expected from the solid work simulation.
- Found many more factors which weigh in for the design as reality varies from simulation.

Version 2 results:

- Fixing of previously discussed issues of version one.
- Making the arm look more pleasing to the eye.
- Proper design and integration of the servo bed and the servos.

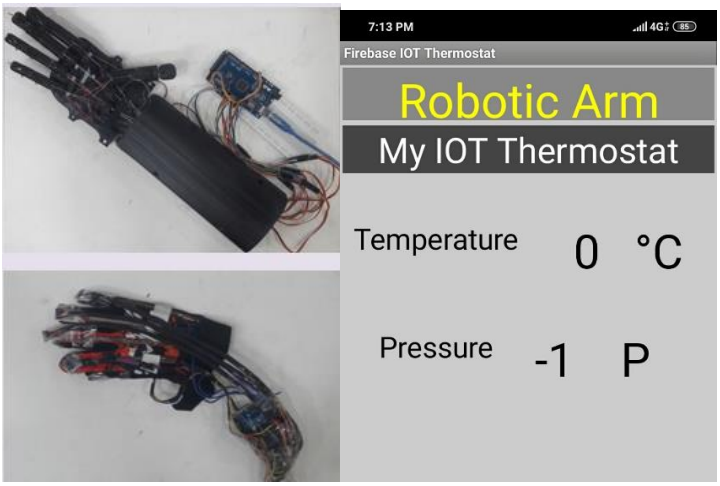

Fig.4.1

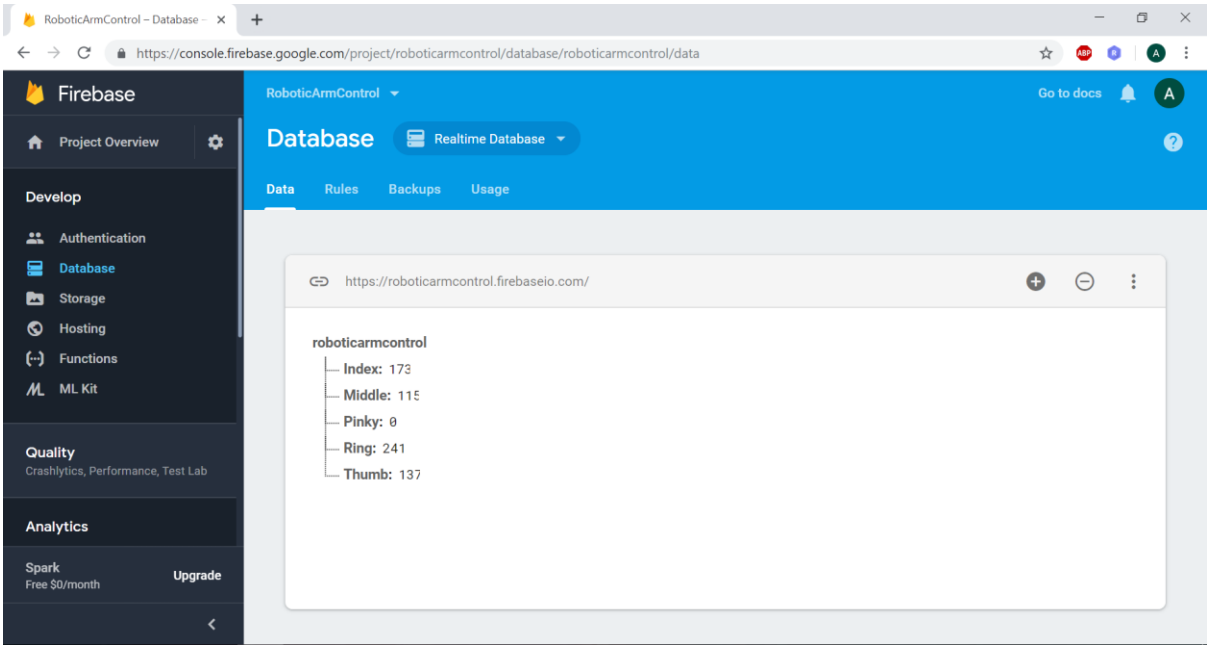

Fig.4.2

The above pictures are the real-time database on the cloud firebase this acts as a place of exchange between the glove and the arm. The Fig 4.1 has a screenshot of the app designed, it can be used to monitor the pressure and the temperature that the arm is currently facing.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

Any Robotic arm has to serve the purpose of being able to fold its fingers and be able to be controlled, in our case the controller is a haptic glove.

5.2 Future Work

There are many directions this project could venture into. Few of our ideas on how we could possibly continue this project are listed below.

The robotic arm which can perform more and more complex motions and make the arm look more appealing to the eye. The design right now was to serve as a proof of concept due to which even though not aesthetically looking good serves our purpose as all the joints were designed after meticulous testing and literature survey and experiments on holding various objects with our own hand.

We would also like to remove the use of the robotic glove so it could be controlled via emg signals produced by the brain. As this would serve as an outstanding prosthetic.

The complexity that this arm can reach is to the level that a doctor should be able to perform complex surgeries on a patient sitting in a whole other country perhaps. For this to be achieved the glove and the arm movements must be perfected as one small slip up could result in possible death. This feature allows doctors to perform more intricate surgeries and also operate on patients on a earlier time by which the rate of survival may shoot up. By reducing the contact of humans during the operation reduces the chance of post-surgery infections for the patients as the robot can be more easily sterilized as compared to the human doctor, this reduces chances of the doctor also contracting dangerous diseases if the patient has a contagious form of disease.

BIBLIOGRAPHY

Papers

[1] Vipul J. Gohil, Dr. S D. Bhagwat, Amey P. Raut, Prateek R. Nirmal "Robotics Arm Control using Haptic Technology" in Mukesh Patel School of Technology & Management Mumbai 56, India

[2] Abidhusain Syed, Zamrrud Taj H. Agasbal, Thimmannagouday Melligeri, Bheemesh Gudur "Flex Sensor Based Robotic Arm Controller Using Micro Controller" in Department of Electronics and Communication, BLDEA College of Enggineering Bijapur-3, India, JSEA> Vol.5 No.5, May 2012

[3] Abhishek Gupta, Student Member, IEEE, and Marcia K. O'Malley, Member, IEEE "Design of a Haptic Arm Exoskeleton for Training and Rehabilitation" in IEEE/ASME TRANSACTIONS ON MECHATRONICS, VOL. 11, NO. 3, JUNE 2006

[4] Waseem Afzal, Shamas Iqbal, Zanib Tahira, Mehtab Ejaz Qureshi "Gesture Control Robotic Arm Using Flex Sensor" in Researchgate Applied and Computational Mathematics 2017;6(4):171-176